Image Sentiment Analysis of Social Media Data

Diandre de Paula

Dissertação para obtenção do Grau de Mestre em
Engenharia Informática
(2º ciclo de estudos)

Supervisor: Prof. Luís Filipe Barbosa de Almeida Alexandre

Covilhã, Julho de 2021

Dedication

Dedicated to my mother, Dinângela Pinto de Paula, who always believed, encouraged me, supported me, and continues to support me at all times, with great effort and dedication, abdicating her life for the cause of her daughters' achievements and happiness.

To my grandfather, Roque Pinto de Paula, who always believed and supported me even though he was far away.

To my sister, Giulianna de Paula, for her care and affection.

Acknowledgements

To God, for allowing me to realize so many dreams in this lifetime and enabling me to continue with my goals. Thank you for allowing me to make mistakes, learn and grow, for Your eternal understanding and tolerance, for Your infinite love, for Your "invisible" voice that allowed me to persist and continue to follow my dreams. Thanks especially for giving me such a special family, anyway, thank you for everything.

To Professor Luís Alexandre, for his guidance, competence, professionalism, and dedication. Although at times I was discouraged and uncertain about my decisions, our meetings were essential in giving me encouragement and motivation to continue. Thank you for believing and trusting me. You were and are being much more than a supervisor: for me, you will always be an example to be followed. I am also grateful for the dedication, professionalism, and passion applied during the classes, which awakened in me, even during the 1st Cycle, the interest in this magnificent and vast area that is Artificial Intelligence.

Special thanks to my mother, for guiding me throughout my life, for the lessons of love, companionship, friendship, charity, dedication, understanding, and forgiveness that you teach me every day. Thank you for being my basis during this journey, for being my friend at all times, for sharing my worries and even my sleepless nights. Without you, I certainly wouldn't be here, and I wouldn't have gotten where I am. There are no words that can express my gratitude for all you have done for me. More than a mother, you are an example to be followed. My fighter mother, loving, dedicated, who takes anything for her daughters. Thank God for presenting me with my mother. I thank God for giving me this special mother, sister, and grandfather. I thank my dear sister, who is always ready to support me in everything in this life, and my grandfather, who always helped me and was proud of me.

To course friends and personal friends, for readings, reviews, questions, and discussions always so productive. And for moments of leisure and support.

Finally, to all those who contributed, directly or indirectly, to the realization of this dissertation, my sincere thanks.

”The beginning of wisdom is the definition of terms.”

Socrates

Resumo

Muitas vezes uma imagem vale mais que mil palavras, e esta é uma pequena afirmação que representa um dos maiores desafios da área de classificação do sentimento contido nas imagens. O principal tema desta dissertação é a realização da análise do sentimento contido em imagens das mídias sociais, principalmente do Twitter, de modo que possam ser identificadas as situações que representam riscos (identificação de situações negativas) ou as quais possam se tornar um (previsão de situações negativas).

Apesar da diversidade de trabalhos feitos na área da análise de sentimento em imagens, ainda é uma tarefa desafiante. Diversos fatores contribuem para a dificuldade , tantos fatores mais globais como questões socioculturais, quanto questões do próprio âmbito de análise de sentimento em imagens, como a dificuldade em achar dados confiáveis e devidamente etiquetados para serem utilizados, quanto fatores enfrentados durante a classificação, como por exemplo, é normal associar imagens com cores mais escuras e pouco brilho à sentimentos negativos, afinal a maioria é assim, entretanto há casos que fogem dessa regra, e são esses casos que afetam a precisão dos modelos desenvolvidos. Porém, visando contornar esses problemas enfrentados na classificação, foi desenvolvido um modelo multitarefas, o qual irá considerar informações globais, áreas salientes nas imagens, expressões faciais de rostos contidos nas imagens e informação textual, de modo que cada componente se complemente durante a classificação.

Durante os experimentos foi possível observar que o uso dos modelos propostos podem trazer vantagens para a classificação do sentimento em imagens e até mesmo contornar alguns problemas evidenciados nos trabalhos já existentes, como por exemplo a ironia do texto.

Assim sendo, este trabalho tem como objetivo apresentar o estado da arte e o estudo realizado, de modo a possibilitar a apresentação e implementação do modelo multitarefas proposto e realização das experiências e discussão dos resultados obtidos, de forma a verificar a eficácia do método proposto. Por fim, as conclusões sobre o trabalho feito e trabalho futuro serão apresentados.

Abstract

Often a picture is worth a thousand words, and this is a small statement that represents one of the biggest challenges in the Image Sentiment Analysis area. The main theme of this dissertation is the Image Sentiment Analysis of social media, mainly from Twitter, so that it is identified as situations that represent risks (identification of negative situations) or that become a risk (prediction of negative situations).
Despite the diversity of work done in the area of image sentiment analysis, it is still a challenging task. Several factors contribute to the difficulty, both more global factors likewise sociocultural issues, and issues within the scope of the analysis of feeling in images, such as the difficulty in finding reliable and properly labeled data to be used, as well as factors faced during the classification, for example, it is normal to associate images with darker colors and low brightness to negative feelings, after all, most are like that, but some cases escape this rule, and it is these cases that affect the accuracy of the developed models. However, in order to overcome these problems faced in classification, a multitasking model was developed, which will consider the entire image information, information from the salient areas in the images, and the facial expressions of faces contained in the images, and textual information, so that each component complements the other during classification.
During the experiments it was possible to observe that the use of the proposed models can bring advantages for the classification of feeling in images and even work around some problems evidenced in existing works, such as the irony of the text.
Therefore, this work aims to present the state of the art and the study carried out, in order to enable the presentation and implementation of the proposed model and carrying out the experiments and discussion of the results obtained, in order to verify the effectiveness of what was proposed. Finally, conclusions about the work done and future work will be presented.

Keywords

Image Sentiment Analysis, Convolutional Neural Network, Multimodal, Image Classification, Dataset, Facial Expression Recognition, Salient Areas, Text Classification.

Contents

List of Figures

List of Tables

Acronyms List

3DIR 3D Inception-ResNet

ACNN Anatomically Constrained Neural Networks

ACPNN Augmented Conditional Probability Neural Network

Adaboost Adaptive Boosting

AIC Affective Image Classification

AMD Advanced Micro Devices

AMP Advanced Multimedia Processing

AMT Amazon Mechanical Turk

ANN Artificial Neural Network

ANP Adjective Noun Pairs

API Application Programming Interface

AUs Action Unions

AutoML Automated Machine Learning

BEs Basic Emotions

BoF Bag of Freebies

BOM Byte Order Mark

BoS Bag of Specials

BU-3DFE Binghamton University 3D Facial Expression

CC Creative Common

CE Compound Emotion

CK+ Extended Cohn–Kanade

CNN Convolutional Neural Network

COCO Commom Objects in Context

CPU Central Process Unit

CSV Comma-Separated Values

DBN Deep Belief Network

DCA Different of Convex functions Algorithm

DCNN Deep Convolutional Neural Network

DMAF Deep Multimodal Attentive Fusion

ESM Emotion Stimuli Map

FACS Facial Action Coding System

FER Facial Expression Recognition

FLOPS Floating Point Operations per Second

FLs Facial Landmarks

FPN Feature Pyramid Network

GAN Generative Adversarial Network

GAPED Geneva Affective Picture Database

GBM Gradient Boosting Machine

GIF Graphic Interchange Format

GPU Graphics Processing Unit

HDD Hard Disk Drive

HOG Histogram of Oriented Gradients

HTML HyperText Markup Language

IAPS International Affective Picture System

IESN Image-Emotion-Social-Net

ILSVRC ImageNet Large Scale Visual Recognition Challenge

IoU Intersection over Union

ISA Image Sentiment Analysis

JAFFE Japanese Female Facial Expressions

JSON JavaScript Object Notation

KDEF Karolinska Directed Emotional Face

kNN k-Nearest Neighbours

LBP Local Binary Pattern

LSTM Long Short-Term Memory

mAP mean Average Precision

MLP Multilayer Perceptron

MMI Maja & Michel Initiative

MSRA Microsoft Research Asia

MTCNN Multi-Task Cascaded Convolutional Neural Networks

Multi-PIE Multi-Pose, Illumination, and Expression

NAFLD Nonalcoholic Fatty Liver Disease

NAPS Nencki Affective Picture System

NASNet-A-Large Neural Architecture Search Network-A-Large

NLP Natural Language Processing

NLTK Natural Language Toolkit

NMS Non-Maximum Suppression

NN Nearest Neighbors

OM Opinion Mining

O-Net Output Network

PCNN Progressive Convolutional Neural Network

PNASNet Progressive Neural Architecture Search Network

P-Net Proposal Network

PNN Probabilistic Neural Network

PP Paddle Paddle

R-CNN Region-based Convolutional Neural Networks

R-FCN Region-based Fully Convolutional Network

R-Net Refinement Network

RAM Random-Access Memory

RAF-DB Real-world Affective Database

RAF-ML Real-world Affective Faces Multi Label

RBM Restricted Boltzmann Machine

ReLU Rectified Linear Activation Function

ResNet Residual Network

RNN Recurrent Neural Networks

ROI Region of Interest

RPN Region Proposal Network

SA Sentiment Analysis

SAM Self-Assessment Manikin

SENet Squeeze-and-Excitation Networks

SFEW Static Facial Expressions in the Wild

SIFT Scale-Invariant Feature Transform

SIMPSoN SocIal MediaPictureS News-related

SPP Spatial Pyramid Pooling

SRC Sparse Representation-based Classifier

SSD Single Shot Detector

SST Stanford Sentiment Treebank

SVM Support Vector Machines

t-SNE t-Distributed Stochastic Neighbor Embedding

T4SA Twitter for Sentiment Analysis

UTF Unicode Transformation Format

V-A Valence-Arousal

VADER Valence Aware Dictionary and sEntiment Reasoner

VDCNN Very Deep Convolutional Networks

VGG Visual Geometry Group

VOC Visual Object Classes

VSO Visual Sentiment Ontology

XML eXtensible Markup Language

YOLO You Only Look Once

ZFNet Zeiler and Fergus Network

Chapter 1

Introduction

This chapter presents the problem statement, as well as the goals to be achieved during the development of this work. Also, it contains the respective dissertation outline.

1.1 Problem Statement

We are increasingly witnessing the growth of the online community, which shows us the different opinions and ways of thinking that each person has. In the same way, as an artist expresses himself through art, users seek ways to express themselves beyond the use of words, often using images to reach their goal. Therefore, the ascent of the use of social media plays a fundamental role, because it's through social media that the users found a place not just to exercise their right of freedom of speech, but also as a news vehicle, which is powerful and can spread easier and quicker. It can also be used as a way of finding and attracting supporters, contestants, and even getting information faster which can be useful to the competent authorities. Thus, social media has posts that pass on to us good feelings and others that will pass on to us not so good feelings. There are many factors to take into account when we need to analyze the sentiment that an image passes to us, for instance, the socio-cultural issues. However, other features can help us to identify the sentiment in the image, for example: prevailing colors in the image, the type of objects in the image, and the metadata (e.g image's caption) that are associated with the image, and through all those factors, we can get a clue to which sentiment the image conveys.
Many works have been done in the image sentiment analysis field. However, there is no way to say that a method is more correct than the other because there are a lot of different ways to approach such a theme. Also, it will depend on the goal to be achieved by the developed model.
Thus, this project aims to develop a model that classifies the image sentiment to identify those images that may represent negative and strongly negative situations since we are interested in predicting when possible strongly negative events are going to take place. This prediction will be obtained not just with the image information from the social media posts, but also with textual information, but that part of the project is done by other team members.

1.2 Objectives

This dissertation has the goal of analyzing the sentiment of images from social media, namely Twitter, which is within the scope of the MOVES project. The collective behavior of crowds can be an agent of social change and an affirmation of existing social norms

and structures, with this the MOVES project proposes to develop a multilingual surveillance system capable of detecting emerging crowds by identifying rising events that foster high focus, high energy and high emotion on social media. Their fundamental hypothesis is that virtual crowds show similar characteristics to real crowds, which may allow their modelization in terms of complex computer systems by relying on advanced natural language processing and machine learning techniques, [mov]. The collective behavior may have dramatic consequences such as crimes and material damages, which badly reflect how fractured our societies can be. Daily reports of protests in various parts of the world emerge, which requires special attention. For this, a model should be employed, which must classify and identify mainly the negative and strongly negative sentiments, which will be also responsible for the image sentiment analysis for the MOVES project.
In this thesis we will create a new model for image sentiment analysis using deep learning methods to be able to classify and identify mainly the negative and strongly negative sentiments by analysing the entire image, the salient areas of the image, the facial expressions (if there is at least one face in the image), and the text that follows the image.

1.3 Dissertation Outline

In order to reflect the work that has been done, this dissertation is organized as follows:

1. The first chapter - **Introduction** - presents the problem statement, the objectives to be accomplished, and the respective dissertation outline;
2. The second chapter - **Related Work** - presents the studies made, discussing related works in the image sentiment analysis field, related techniques, and available approaches;
3. The third chapter - **Proposed Method and Implementation** - presents an overview of the proposed model, presenting in detail its components;
4. The fourth chapter - **Results and Discussion** - presents the experiments that were made and the results obtained, and a discussion about the values and behaviour observed from the tests;
5. The fifth chapter - **Conclusion** - contains the main conclusions about the work that has been done. Also, it presents the contributions to be achieved, and the future work.

Chapter 2

Related Work

2.1 Introduction

This chapter presents the most relevant work in the image sentiment analysis field, as well as the most common datasets and their features. This chapter is split as follows: section 2.2 presents the used and available datasets in the image sentiment analysis field. The following section 2.3 presents the approaches that can be used in the image sentiment analysis field. Section 2.4 presents the tools and technologies with object detection that we can use. Section 2.5 presents the approaches used in the facial expression recognition field. Section 2.6 presents the sentiment models that are used in the image sentiment analysis field. Section 2.7 analyses the works on image sentiment analysis, and the final section 2.8 contains the main conclusions made while formulating this chapter.

2.2 Datasets

This subsection presents the available datasets and an overview about them. It also presents the datasets used in the studied works and the results obtained.

2.2.0.1 International Affective Picture System (IAPS) - 1999

In 1997 an intensive study was performed on the IAPS in order to extract a categorical structure of such dataset [BL17]. As a result, a database of photos that have been validated as consistently eliciting a specific emotional response in viewers was obtained. Being a psychological dataset, it's very difficult to be built in large scale and maintained over time [OFB20]. According to [OFB20], the IAPS dataset is composed by 716 photos, which are not from social media. Also, it is not classified into polarity nor contains any additional metadata as text. To obtain this dataset, a formal request is necessary [BL].

2.2.0.2 AIC Using Features inspired by Psychology and Art Theory - 2010

In [MH10b] methods were developed to extract and combine low-level features that represent the emotional content of an image, and use these for image emotion classification. For testing and training, they used three datasets:

- IAPS (I);
- A set of artistic photography from a photo sharing site (II);
- A set of peer rated abstract paintings (III).

The latter two datasets (II and III) were collected for their study, and they are available to the research community [MH10a]. According to the [OFB20] the datasets are composed by 228 paintings and 807 photos, and they have the following categories: awe, amusement, contentment, excitement, disgust, anger, fear, and sadness. However, the photos are not from social media and the dataset did not contain additional metadata.
Figure 2.1 shows an example of the photo contained in the dataset (II).

Figure 2.1: Example of a photo contained in the AIC dataset (II), which represents excitement. (Source: image from the dataset available on [MH10a]).

Figure 2.2 shows an example of the painting contained in the dataset (III).

2.2.0.3 Geneva Affective Picture Database (GAPED) - 2011

GAPED is a database consisting of 730 photos, and it was created to increase the availability of visual emotion stimuli. Four specific negative contents were chosen: spiders, snakes, and scenes that induce emotions related to violation of moral and legal norms (human rights violation or animal mistreatment). Positive and neutral pictures were also included: positive pictures represent mainly human and animal babies as well as nature scenarios, while neutral pictures mainly depict inanimate objects. The pictures were rated according to valence, arousal, and the congruence of the represented scene with internal (moral) and external (legal) norms [dG]. According to the [OFB20] this dataset is used to classify the images into positive, negative, and neutral, and it is a psychological dataset. There is no additional metadata and the images are not from social media. The dataset can be downloaded from [dG], which has 402 MB.

2.2.0.4 VSO - 2013

Visual Sentiment Ontology (VSO) is the largest benchmark dataset for visual sentiment prediction, which has about 1.4 million images from 3,244 Adjective Noun Pairs (ANP).

Figure 2.2: Example of a painting contained in the AIC dataset (III), which have the following groundtruth: amusement - 0, anger - 0, awe - 1, content - 2, disgust - 0, excitement -0, fear - 2, sad - 3. (Source: image from the dataset available on [MH10a]).

This dataset is collected by querying Flickr with ANPs. There are a total of 269 adjectives (attributes), which are considered to be sentiment related. Among them, 127 attributes are labeled as positive and the others are negative. Each image is associated with one ANP. Thus, each image is labelled according to the sentiment label of its ANP [WMW20]. In [DBC] there are several datasets for download, which are:

- **VSO - Ontology and Concepts (200.88 KB):**
  - **Analysed Images & Videos:** 316,000;
  - **Non-empty ANP candidates:** 47,000;
  - **VSO ANP:** 3,244;
  - **ANP included in SentiBank:** 1,200.
- **VSO - Image Dataset (58.23 GB):** the database consist of two datasets, a set of Flickr images with Creative Common (CC) licenses used in training/testing 1,200 ANPs detectors in SentiBank and the set of images associated with the full VSO including 3,244 ANPs;
- **SentiBank - Visual Sentiment Concept Classifiers (975 MB):** this dataset was used to train the concept detector. For each ANP a concept detector has been trained using 80% of its Flickr sample images. The remaining 20% have been used for detector testing;

- **Photo Tweet Sentiment Benchmark (56.92 MB):** the benchmark includes 603 tweets with photos and is intended for evaluating the performance of automatic sentiment prediction using features of different modalities (text only, image only, and text-image combined). It was collected in November 2012 via the PeopleBrowsr Application Programming Interface (API) using 21 hashtags. The groundtruths of sentiment values were obtained by Amazon Mechanic Turk annotation, resulting in 470 positive and 133 negative labels.

Figure 2.3 shows an example of the images contained in the Photo Tweet Sentiment Benchmark dataset.

Figure 2.3: Example of an image contained in the VSO dataset. (Source: image from the dataset available on [DBC]).

The VSO dataset contains additional metadata, for example, the Figure 2.4 presents the additional metadata of the Figure 2.3

2.2.0.5 Nencki Affective Picture System (NAPS) - 2014

Another psychological dataset [OFB20], the NAPS consists of 1,356 realistic, high-quality photographs that are divided into five categories (people, faces, animals, objects, and landscapes). Affective ratings were collected from 204 mostly European participants. The pictures were rated according to valence, arousal, and approach-avoidance dimensions using computerized bipolar semantic slider scales. Validation of the ratings was obtained by comparing them to ratings generated using the SAM and the IAPS. In addition, physical properties of the photographs are reported, including luminance, contrast, and entropy. The new database, with accompanying ratings and image parameters, allows researchers to select a variety of visual stimulus materials specific to their experimental questions of interest [MZJG14], [Mat]. To download the dataset, it is necessary to fill out a request form [LOB14].

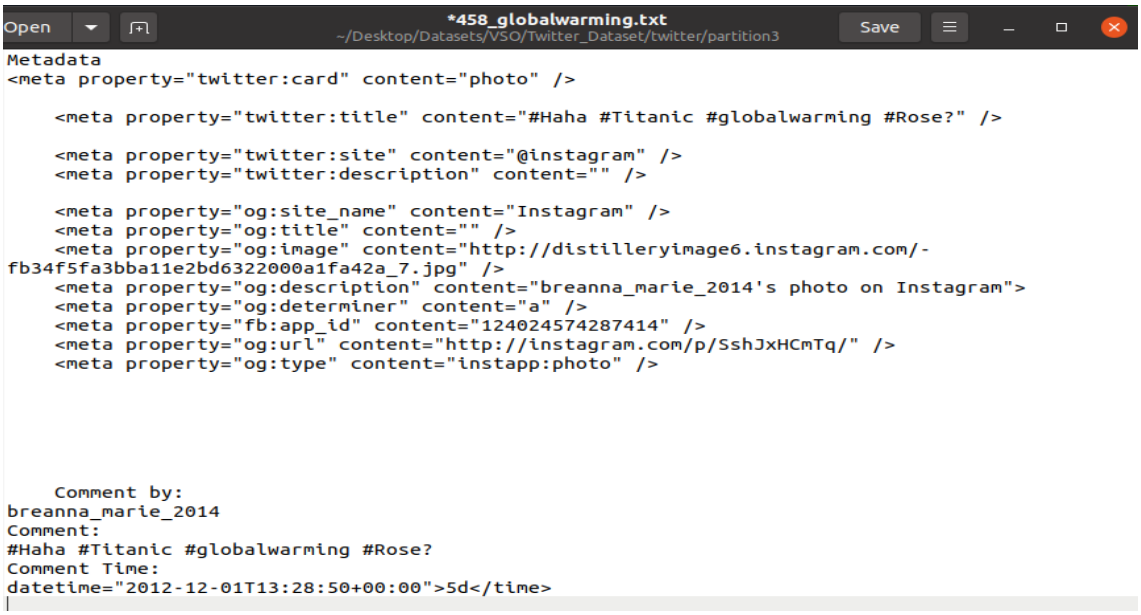


Figure 2.4: Additional metadata of the Figure 2.3, which can be found in the VSO dataset.

2.2.0.6 Emotion6 - 2015

The work [PCSG15] explores two new aspects of photos and human emotions. The authors presented a new database, Emotion6, containing distributions of emotions. The dataset consists of 1,980 images collected from Flickr by using the emotion keywords and synonyms as search terms. There are 330 images for each emotion category. Amazon Mechanical Turk (AMT) workers were invited to label the images into the Ekman's 6 emotions and neutral to obtain the emotional responses. Each image was scored by 15 subjects.The discrete emotion distribution information is released. The considered emotions are: anger, disgust, fear, joy, sadness, and surprise. Also, the dataset contains the Valence and Arousal scores.
Figure 2.5 shows an example of the images contained in the Photo Tweet Sentiment Benchmark dataset.
The dataset can be downloaded (181 MB) on Advanced Multimedia Processing (AMP) Lab, Cornell University website [Sad].

2.2.0.7 Image-Emotion-Social-Net (IESN) - 2016

The IESN dataset is constructed for personalized emotion prediction [ZYG+16a], [ZYG+16b], with 1,012,901 images from Flickr. Lexicon-based methods are used to segment the text of metadata from uploaders for expected emotions and comments from viewers for actual emotions. Synonym based searching is employed to obtain the Mikels' emotion category by selecting the most frequent synonyms [ZDH+18].
In [Zha] it is possible to find two datasets:

- **IESN_V1.0:** related to the work [ZYG+16a], this is the data of IESN dataset, which is designed for various visual emotion analysis tasks. It contains 358 MB of JavaScript Object Notation (JSON) files with the following characteristics:

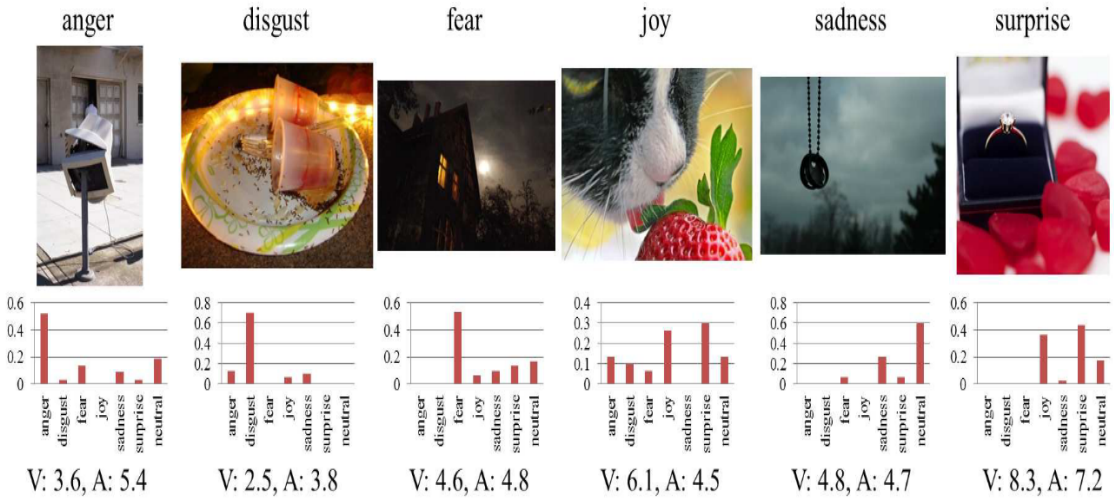


Figure 2.5: Example images of Emotion6 dataset with the corresponding ground truth. The emotion keyword used to search each image is displayed on the top. The graph below each image shows the probability distribution of evoked emotions of that image. The bottom two numbers are V-A scores in SAM 9-point scale. (Source: image from [PCSG15]).

  - **ExpectedEmotion:** Emotions from the image uploaders obtained by the title, tags and descriptions;
  - **ActualEmotion:** Emotions from image comments;
  - **Groups:** Flickr interest groups;
  - **Users:** user information (image uploaders and commenters).

- **IESN_Continuous Distribution_V1.0**: related to the work [ZYG+16b], this is the data used for probability distribution modeling of image emotions. It contains 2.1 GB of images and 11 MB of the respective valence-arousal labels.

2.2.0.8 FI - 2016

The original FI dataset consists of 90,000 noisy images collected from Flickr and Instagram by searching the emotion keywords [KS16]. The weakly labeled images are further labeled by 225 AMT workers, which are selected through a qualification test. The 23,308 images that receive at least three votes from their assigned 5 AMT workers are kept. The number of images in each Mikels' emotion category is larger than 1,000.
Table 2.1 shows an overview about the statistics of FI dataset.

DataSet	Amusement	Anger	Awe	Contentment	Disgust	Excitement	Fear	Sadness	Sum
Submitted	11,000	11,000	11,000	11,000	11,000	11,000	13,000	11,000	90,000
Labeled	4,942	1,266	3,151	5,374	1,658	2,963	1,032	2,922	23,308

Table 2.1: Table presenting the statistics of the current labeled images in FI dataset. (Source: image from [KS16]).

2.2.0.9 EmotionROI - 2016

In [PSGC16] the authors built a dataset as a benchmark for predicting the Emotion Stimuli Map (ESM), which describes pixel-wise contribution to evoked emotions. The authors used images in the Emotion6 dataset to reach their goal. The EmotionROI database contains the ground truth ESMs collected by asking people to identify the regions in the images which most influence their evoked emotions. Thus, the authors used the AMT to collect responses from subjects, building the ground truth ESMs in EmotionROI. Also, they kept the categories used in Emotion6 dataset and created 220 different AMT tasks (each one contains 10 images) for AMT that meet the following constraints:

1. Each AMT task contains at least one image from each of the 6 categories;
2. Images are ordered in such a way that the frequency of an image from category i appearing after category j is equal for all i, j.

The authors enforce the following regulations to be consistent with the Emotion6 database:

1. The same subject can only respond to each image or AMT task at most once, and each subject cannot respond to more than 55 different AMT tasks to increase diversity;
2. 15 responses was collected for each image to have statistically significant results.

The ground truth ESMs was normalized to the range between 0 to 1. Figure 2.6 shows some example images in EmotionROI and the corresponding ground truth ESMs. The dataset is openly and available (195MB) on AMP Lab, Cornell University website [Sad].

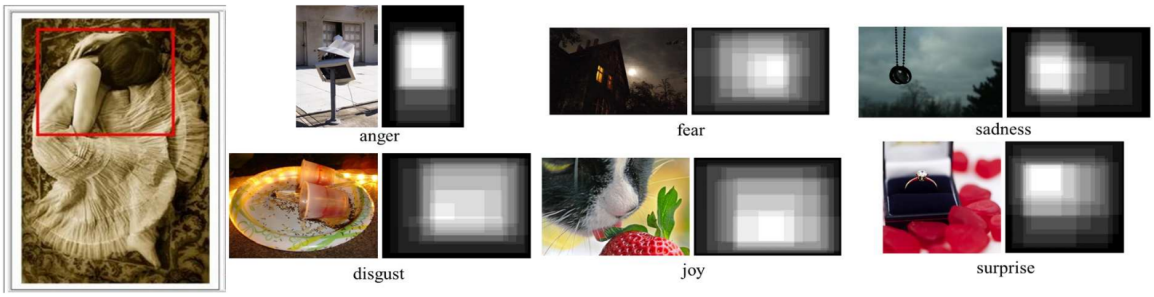


Figure 2.6: The leftmost image is a screenshot of the interface of their user study on Amazon Mechanical Turk. The other images are some examples from EmotionROI dataset with the corresponding ground truth emotion stimuli maps. The emotion keyword used to search each image (provided by Emotion6 dataset [PCSG15]) is displayed under the image. (Source: image from [PSGC16]).

2.2.0.10 T4SA & B-T4SA - 2017

In [VCC+17a], the authors trained a model for visual sentiment classification starting from a large set of user-generated and unlabeled contents. They collected more than 3 million tweets containing both text and images. The authors used the Twitter's Sample API to access a random 1% sample of the stream of all globally produced tweets, discarding:

- Tweets not containing any static image or containing other media (i.e., they also discarded tweets containing only videos and/or animated Graphic Interchange Format (GIF)s);
- Tweets not written in the English language;
- Tweets whose text was less than 5 words long;
- Retweets.

At the end of the data collection process, the total number of tweets in T4SA dataset was about 3.4 million, corresponding to approximately 4 million images. Each tweet (text and associated images) has been labeled according to the sentiment polarity of the text (negative = 0, neutral = 1, positive = 2) predicted by our tandem Long Short-Term Memory (LSTM)-Support Vector Machines (SVM) architecture, obtaining a labeled set of tweets and images divided in 3 categories. The authors of the tweets having the most confident textual sentiment predictions to build their T4SA dataset. The corrupted and near-duplicate images have been removed, and they selected a balanced subset of images, named B-T4SA, that was used to train their visual classifiers.
Table 2.2 shows the details of the dataset.

Sentiment	T4SA (tweets)	T4SA (images)	T4SA (w/o near dup - images)	B-T4SA
Positive	371,341	501,037	372,904	156,862
Neutral	629,566	757,895	444,287	156,862
Negative	179,050	214,462	156,862	156,862
Sum	1,179,957	1,473,394	974,053	470,586

Table 2.2: Table presenting T4SA dataset information.

It's possible to get access to the T4SA dataset through filling a request form available in their website [Vad].
The presented datasets are the most used in the Image Sentiment Analysis field. However, it's possible to find other datasets, in platforms like Kaggle, that can be used to train models to execute this task. For example, the dataset available on [Hsa], which has about 32,000 Flickr images. Table 2.3 summarizes the dataset used in some of the works presented in section 2.7.
Table 2.4 presents the respective results obtained from each work presented in Table 2.3.
Choosing datasets to train a model can be a tough task. Thus, it's necessary to filter the requirements to downsize the number of available options. Aiming the Image Sentiment Analysis for social media images, we can discard (at first) psychological datasets based, due on their number of images. Thus, datasets like B-T4SA, VSO, Flickr, IESN can be good choices due to the huge amount of data contained.
Also, an approach to be investigated is the use of facial expressions. In all studied works none has presented the idea of evaluating the sentiment/polarity taken into account the

Work	Flickr [BJC+13]	FI [KS16]	Twitter I [YLJY15a]	Twitter II [BJC+13]	B-T4SA [VCC+17a]
[YLJY15b]	X				
[GA19a]					X
[GA19b]					X
[ZWS+20]		X	X		
[WQJZ20a]	X	X	X	X	
[FCd19]	X	X			
[VCC+17b]					X

Table 2.3: Studied works and the datasets used.

Work	Flickr [BJC+13]	FI [KS16]	Twitter I [YLJY15a]	Twitter II [BJC+13]	B-T4SA [VCC+17a]	EmotionROI [PCSG15]
[YLJY15b]	0.7730					
[GA19b]					**0.5234**	
[ZWS+20]		0.7572	**0.8715**			
[WQJZ20a]	0.7239	**0.8884**	0.8604	**0.8097**		**0.8304**
[FCd19]	**0.9159**	0.8635				
[VCC+17b]						0.5130

Table 2.4: Results obtained from the studied works for each dataset used.

classification of the facial expressions in images that contain faces. Thus, there are two options to approach this idea:

- Use a pre-trained model;
- Find a dataset and train a model to classify facial expressions.

However, not all images on social media contain faces. Thus, it could be a waste of effort to find a dataset and train a model to execute this task, and the idea of using a pre-trained model seems to be the best.
As already seem through the studied works, the evaluation of local and global images is also crucial. Thus, this approach will be included in the model. The challenge is to decide which approach to be taken. It's hard to say which one is better when we have many possibilities. We can have:

1. Evaluate polarity in local and global images;
2. Evaluate sentiment in local and global images;
3. Evaluate polarity in local images and sentiment in global images;
4. Evaluate sentiment in local images and polarity in global images.

The use of the text available also seems interesting, because can help the model to classify correctly. However, as seen at [FCd19] the analysis of the text is worth only if there is metadata available and the use of a text classification algorithm presented a similar accuracy when compared with a model trained from scratch. So to analyze the metadata the use of a text classification algorithm seems to be more advantageous.

2.3 Traditional Machine Learning Algorithms and Deep Learning Models

With the variety of approaches that can be taken, mainly due to the research made, before making the decision about which approach to use, it was necessary to understand the difference between them.
Traditional machine learning is a branch of artificial intelligence where engineers and scientists manually select features within the data and train the model [DeL].
Deep learning is a branch of machine learning modeled loosely on the neural pathways of the human brain where the algorithm automatically learns what features are useful.

- **Traditional machine learning:** typically used for projects that involve predicting output or uncovering trends. A limited body of data is used to help the machines learn patterns that they can later use to make a correct determination on new input data. Some examples of traditional machine learning algorithms are: linear/logistic regression, decision trees, SVM, naive Bayes, and discriminant analysis;
- **Deep learning:** typically used for projects that involve classifying images, identifying objects in images, and enhancing images and signals. They are designed to automatically extract features from spatially and temporally organized data such as images and signals. Some examples of deep learning methods are: Convolutional Neural Network (CNN)s, Recurrent Neural Networks (RNN), and reinforcement learning (deep Q networks).

Table 2.11 shows some considerations to take into account before choosing an approach.

Consideration	Traditional Machine Learning	Deep Learning
Data consideration	Available data is more limited and structured	Requires a large quantity of training data
Available Hardware and Deployment	Require less computational power	Require specialized hardware (GPUs)

Table 2.5: Results obtained from the studied works for each dataset used.

Thus, traditional machine learning algorithms may be more desirable if you need quicker results. They are faster to train and require less computational power. The number of features and observations will be the key factors that affect training time. With traditional machine learning, it is expected to spend the majority of time developing and evaluating features to improve model accuracy.
Deep learning models will take more time to train, but pre-trained networks and public datasets can shorten training through transfer learning. With deep learning, it is expected to spend a majority of time training models and making modifications to the architecture of deep neural networks. However, some works investigated the use of both approaches, as seen in [ZWS+20].

2.3.0.1 Transfer Learning

Transfer learning is a popular method in computer vision because it allows us to build accurate models in a time-saving way [RW17]. With transfer learning, instead of starting the learning process from the scratch, you start from models that have been trained on a different problem.
Transfer learning usually uses pre-trained models, which are models that were trained on a large benchmark dataset to solve a problem similar to the one that we want to solve.
Several pre-trained models used in transfer learning are based on a large CNN [VDDP18].
A typical CNN has two parts:

- **Convolutional base:** the main goal is to generate features from the image;
- **Classifier:** the main goal is to classify the image based on the detected features.

To repurpose a pre-trained model for your own needs, you start by removing the original classifier, then you add a new classifier that fits your purposes. Then, it is necessary to fine-tune the model according to one of three strategies:

1. Train the entire model;
2. Train some layers and leave the others frozen;
3. Freeze the convolutional base.

Figure 2.7 shows these three strategies schematically.

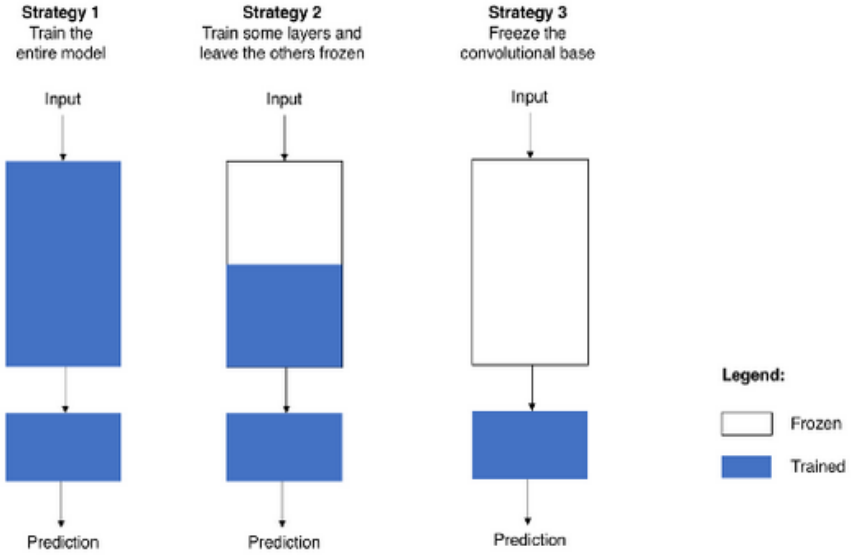


Figure 2.7: Fine-tuning strategies. (Source: image from [Mar]).

When you're using a pre-trained model based on a CNN (except Strategy 3), it's smart to use a small learning rate because high learning rates increase the risk of losing previous knowledge.
The process of transfer learning can be divided into:

1. Select a pre-trained model.
2. Classify your problem according to the Size-Similarity Matrix (Figure 2.8 – left).
3. Fine-tune your model (Figure 2.8 - right).

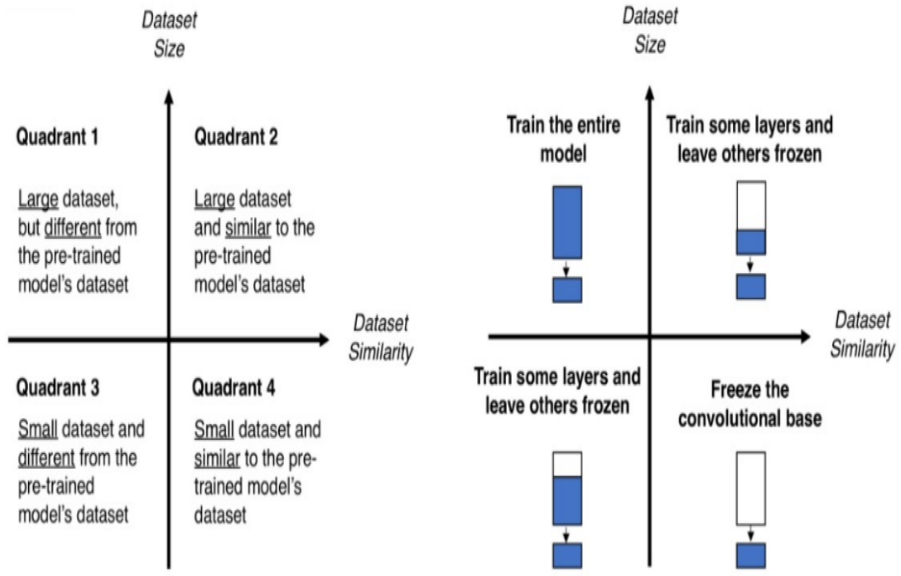


Figure 2.8: Size-Similarity matrix (left) and decision map for fine-tuning pre-trained models (right). (Source: image from [Mar]).

The work [Li] aims to summarize the study made in [KSL19]. As seen, the idea of transfer learning is adapting the model trained with a big dataset to your problem. There are two types of transfer-learning: **i)** use fixed features; **ii)** fine-tuning. Both involve restoring weights from a pre-trained ImageNet model and retraining the network for the new classes of interest. The difference between both of them is:

- **Fixed features approach:** freezes early layers and only trains the last layer;
- **Fine-tuning approach:** trains all layers.

The fixed features approach is less prone to overfitting, while the fine-tuning approach is better at handling new classes. Google researchers set themselves the goal of studying the pros and cons of these two approaches. Figure 2.9 shows the results of their experiments. We can see if we have a large dataset but the data is different from the pre-trained model's dataset, it will be necessary to train the entire model. For example, the pre-trained model VGGNet couldn't help if the goal of the task is to predict over renal biopsy because there are no images like this in the ImageNet dataset.
From the experiments, the researchers could find:

- Datasets similar to ImageNet benefit more from transfer learning than dataset un-similar to ImageNet;
- Both fine-tuning and trained from scratch led to significantly better features;

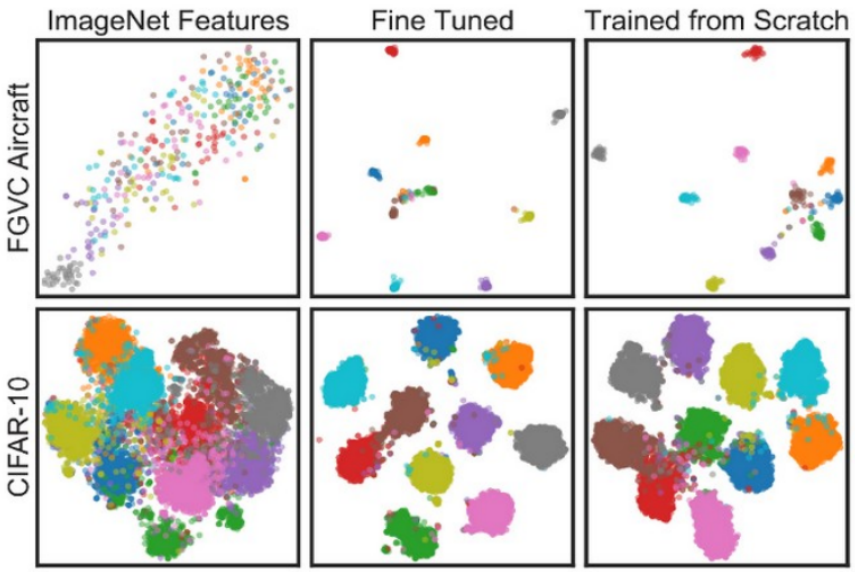


Figure 2.9: 2D embedding t-SNE of features from the penultimate layer of Inception for different approaches. (Source: image from [VDDP18]).

- Although training from scratch achieved parity with fine-tuning, it is at the cost of significantly more training data and longer training time.

Also, the researchers concluded that ResNet seems to be the best fixed features extractor. However, their affirmation is completely empirical-based and there is no theory behind it. The authors of the review suggest trying at least three pre-trained models, which are: ResNet, Inception-V4, and Neural Architecture Search Network-A-Large (NASNet-A-Large). However, among twelve transfer tasks, the best model on ImageNet is NASNet-A-Large. It has won 9 of the 12 transfer tasks in the study. Thus, for fine-tuning, NASNet-A-Large could be a good choice. Thus, once it is relatively easy and does not require too much time (depending on the purpose), a good strategy would start with pre-trained models and evaluate their accuracies, then use a state-of-art dataset large enough to train a model from the scratch.

2.3.0.2 VGG

There are a total of six VGGNet architectures. VGG-16 and VGG-19 are the most popular [Bas]. Every VGG architecture has filters of size 3 x 3, because two 3 x 3 filters almost cover what a 5 x 5 filter would cover. Also, two 3 x 3 filters are cheaper (total number of multiplications to be performed) than one 5 x 5 filter. However, VGGNet has a problem that is this naive architecture is not good for a deeper network – as the network goes deeper, it is more prone to the vanishing gradients' problem, which occurs when the calculated partial derivatives, used to compute the gradient, go deeper into the network. Since the gradients regulate how much the network learns during training, if the gradients are very small or zero, then little to no training can take place, attending to poor predictive performance. More training and also more parameters have to be tuned in deeper VGG architectures.

But VGGNets are handy for transfer learning and small classification tasks. Figure 2.10 shows the VGGNet-16 architecture.

Figure 2.10: VGGNet-16 architecture. (Source: image from [Bas]).

2.3.0.3 ResNet

The success recipe of ResNet for training a deep (152 layers) network is that it has residual connections [Bas]. Figure 2.11 shows the ResNet architecture.

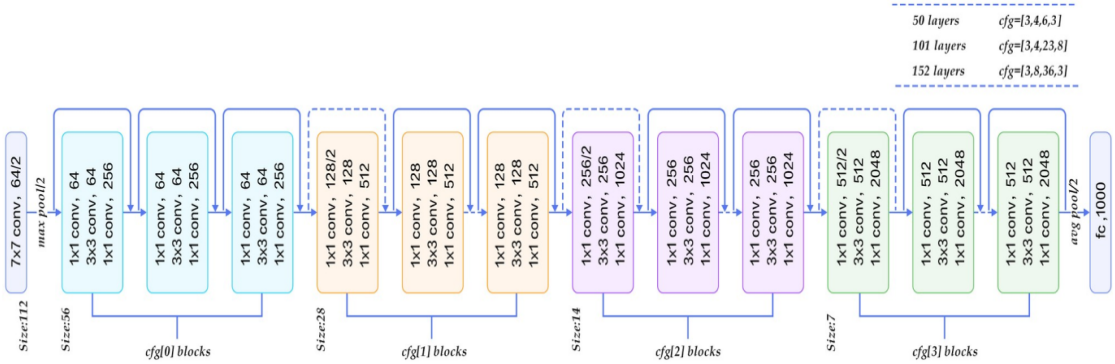


Figure 2.11: ResNet architecture. (Source: image from [Bas]).

In VGGNet, every layer is connected to its previous layer, from which it is getting its input (this makes sure that useful features are propagated, and the less important features are dropped out). However, the latter layers can not see what the former layers have seen. ResNet address this problem by connecting not just the previous layer to the current one, but also a layer behind the previous layer. Training such a deep residual network is possible by using batch normalisation layers after every convolutional layer. These layers will boost the values of weights and hence higher learning rates can be used while training (will help train faster and can minimize the vanishing gradient problem).

2.3.0.4 DenseNet

In this architecture, proposed in [HLvdMW18], for a given layer, all other layers preceding it are concatenated, and given as input to the current layer. Figure 2.12 shows DenseNet's

architecture. Smaller filters counts can be used to minimize the vanishing gradient problem as all layers are directly connected to the output, and gradients can be calculated directly from the output for each layer [Bas].

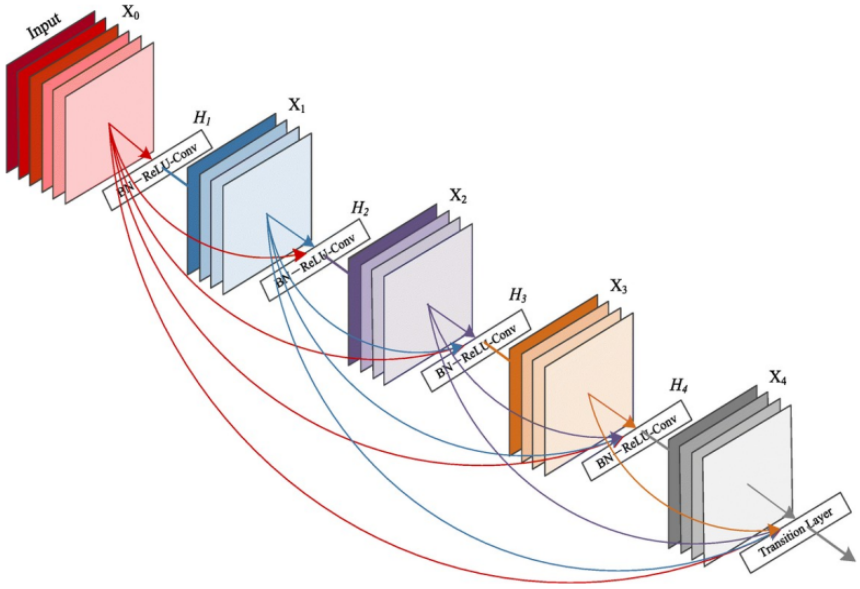


Figure 2.12: DenseNet architecture. (Source: image from [HLvdMW18]).

2.3.0.5 InceptionNet

In ResNet, the focus is on deeper networks. The idea of InceptionNet is to make the network wider. This can be done by parallel connection of multiple layers having different filters and then finally concatenating all of those parallel paths to pass to the next layers [Bas]. Figure 2.13 shows the InceptionNet architecture.

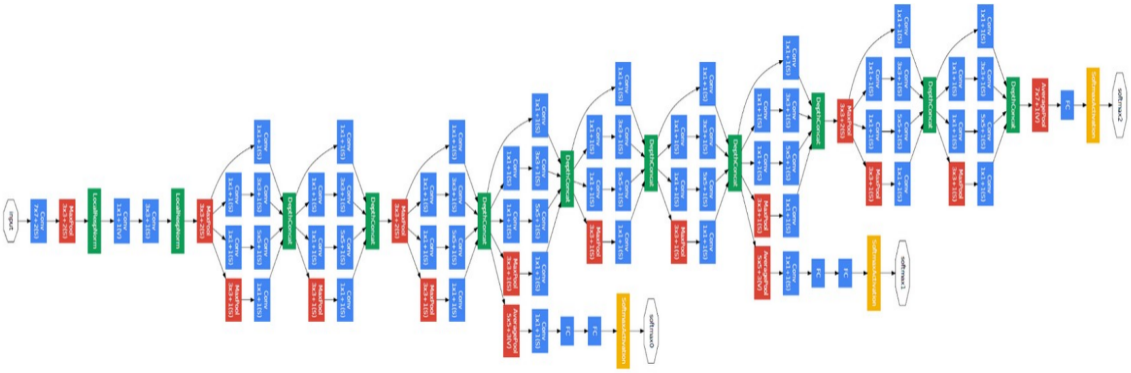


Figure 2.13: InceptionNet architecture (Source: image from [Bas]).

There are many variants of InceptionNets. The differences among them are:

- Instead of using a 5 x 5 filter, use two 3 x 3 filters as they are computationally efficient;
- Using a 1 x 1 Conv2D layer with smaller filter count before performing any Conv2D

layer with large filter sizes will reduce the depth of the input and hence is computationally efficient;

- Instead of performing a 3 x 3 filter, perform a 1 x 3 filter followed by a 3x1 filter. This will drastically improve the computational efficiency.

Table 2.6 shows the winners of the ILSVRC [Sha], [Siy].

Model	Error Rate (%)	Year
AlexNet (SuperVision)	15.3	2012
Zeiler and Fergus Network (ZFNet) (Clarifai)	11.2	2013
Inception (GoogLeNet)	6.67	2014
VGG-16 (Runners-Up)	7.3	2014
ResNet (Microsoft Research Asia (MSRA))	3.57	2015
ResNetXt-10	4.1	2016
Squeeze-and-Excitation Networks (SENet)	2.251	2017
Progressive Neural Architecture Search Network (PNASNet)-5	3.8	2018

Table 2.6: Winners of ILSVRC competition.

In [SP18], a comparison between AlexNet, VGG-16, and VGG-19, to transfer learning was made. The authors concluded that when using VGG-19 CNN architecture there was an improvement in the average recall, precision, and F-score on both the databases, CalTech256 [GHP07], and GHIM10K [LW03].
In the work [MGGTZC+20] we can find a comparison between AlexNet, GoogleNet, InceptionV3, ResNet18, and ResNet50 for the classification of tomato plant diseases. The authors found that every model used in their work was capable of classifying nine diseases in tomato leaves from the healthy class, where the GoogleNet model with 22 layers can reach 99.72% classification of tomato diseases using the training mechanism of transfer learning. On the other hand, Inception V3 obtained the lowest performance compared to the other architectures.
In the work [ACT+20], a comparison between a custom CNN (CNN L_Rectified Linear Activation Function (ReLU) topology) and pre-trained models (AlexNet, VGG-19, GoogleNet, and InceptionV3) was made for Nonalcoholic Fatty Liver Disease (NAFLD) Biopsy Images. The results showed that the custom CNN achieved a 95.8% classification accuracy, while AlexNet produced the highest classification performance (accuracy:97.8%).
In the work [GA19a] a comparison between ResNet18, ResNet50, ResNet152, InceptionV3, and DenseNet161 was made to improve the state of the art in a large tweet data set. The authors found that it was possible to improve the accuracy value with DenseNet (52.74%), with an increase of 0.76% from the previous work [GA19b]. However, looking towards the execution time presented, it's possible to conclude that ResNet50 reaches a similar value (52,51%), but in a shorter time, 78 hours and 17 hours, respectively.

2.4 Object Detectors

According to the studied papers, it's almost mandatory to conclude that it is important to analyze the sentiment in the salient region of the image, and take into account the objects

in the image. Thus, we need to analyze the available algorithms to make object detection. The object detection task involves object classification and object localization, and both are challenging topics in the domain of computer vision.
The source [Cho20] presents eight algorithms for object detection, which are:

- Histogram of Oriented Gradients (HOG);
- Spatial Pyramid Pooling (SPP)-net;
- Region-based Convolutional Neural Networks (R-CNN);
- Fast R-CNN;
- Faster R-CNN;
- Region-based Fully Convolutional Network (R-FCN);
- Single Shot Detector (SSD);
- YOLO.

The author presented a description of each one based on their respective works. Thus, a brief description will be presented below.

2.4.0.1 HOG (2005)

HOG is a feature extractor that can be used to detect objects in image processing and other computer vision techniques [DT05]. The HOG descriptor technique includes occurrences of gradient orientation in localized portions of an image, such as the detection window, the ROI, etc.

2.4.0.2 SPP-net (2014)

SPP-net is a network structure that can generate a fixed-length representation regardless of image size/scale [HZRS14]. This method avoids repeatedly computing the convolutional features.

2.4.0.3 R-CNN (2014)

The R-CNN is a combination of region proposals with CNNs. It helps in localizing objects with a deep network and training a high-capacity model with only a small quantity of annotated detection data [GDDM14]. R-CNN can scale to thousands of object classes without resorting to approximate techniques including hashing.

2.4.0.4 Fast R-CNN (2015)

This algorithm mainly fixes the disadvantages of R-CNN and SPP-net, while improving their speed and accuracy, [Gir15].
Advantages:

- Higher detection quality (mAP) than R-CNN, SPP-net;
- Training is single-stage, using a multi-task loss;
- Training can update all network layers;
- No disk storage is required for feature caching.

2.4.0.5 Faster R-CNN (2015)

This algorithm utilizes the Region Proposal Network (RPN) that shares full-image convolutional features with the detection network in a more cost-effective manner than R-CNN and Fast R-CNN, [RHGS15]. The RPN is a fully convolutional network that simultaneously predicts the object bounds as well as the score at each position of the object and is trained end-to-end to generate high-quality region proposals.

2.4.0.6 R-FCN (2016)

The R-FCN is a region-based detector for object detection. It's fully convolutional with almost all computation shared on the entire image, [DLHS16]. In this algorithm, all learnable weight layers are convolutional and are designed to classify the ROIs into object categories and backgrounds.

2.4.0.7 SSD (2016)

The SSD is a method for detecting objects in images using a single deep neural network. The SSD approach discretizes the output space of bounding boxes into a set of default boxes over different aspect ratios, [LAE+16]. It combines prediction from multiple feature maps with different resolutions to naturally handle objects of various sizes.
Advantages:

- Eliminates proposal generation and subsequent pixel or feature resampling stages and encapsulates all computation in a single network;
- Easy to train and to integrate into systems.

2.4.0.8 YOLO (2016)

YOLO is one of the most popular algorithms in object detection used by researchers, [RDGF16]. The base YOLO model processes images in real-time at 45 frames per second while a smaller version of the network, Fast YOLO, processes 155 frames per second while still achieving double mAP of other real-time detectors.
The first three YOLO models have been released between 2016 and 2018. However, in 2020, three major versions of YOLO have been released, which are: YOLOv4, YOLOv5, and Paddle Paddle (PP)-YOLO.

- **YOLOv4:** released in April 2020, takes the influence of the state of art Bag of Freebies (BoF) and Bag of Specials (BoS). The BoF improves the accuracy of the detector, without increasing the inference time. YOLOv4 is based on the Darknet;
- **YOLOv5:** released in June 2020, YOLOv5 is different from all other prior releases, as this is a PyTorch implementation and the major improvements include mosaic data augmentation and auto-learning bounding box anchors;
- **PP-YOLO:** introduced in July 2020, PP-YOLO is based on PaddlePaddle (Parallel Distributed Deep Learning), an open source deep learning platform. Its developers aimed to implement an object detector with relatively balanced effectiveness and efficiency that can be directly applied in actual application scenarios.

In [Hui18] the author did a comparison between Faster R-CNN, R-FCN, SSD, Feature Pyramid Network (FPN), RetinaNet, and YOLOv3. The speed and accuracy were evaluated. The author affirms that is very hard to have a fair comparison among different object detectors and that there is no straight answer on which model is the best. He also points to the need to be aware of other choices that impact the performance, for instance:

- Features extractors (VGG, ResNet, Inception, MobileNet);
- Output strides for the extractor;
- Input image resolutions;
- Matching strategy and Intersection over Union (IoU) threshold (how predictions are excluded in calculating loss);
- The number of proposals or predictions;
- Boundary box encoding;
- Data augmentation;
- Training dataset;
- Use of multi-scale images in training or testing.

In his study, the author summarizes the results from individual papers. Due to the difficulty to compare results from different papers, whose experiments were done with different settings, the author plotted them together to make it easier for the reader to have an idea about their performance. The results were obtained from the following settings:

- **Data training:** PASCAL VOC 2007 and 2012;
- **mAP measurement:** PASCAL VOC 2012 testing set;
- **SSD:** 300 x 300 and 512 x 512 input images;
- **YOLO:** 288 x 288, 416 x 416, and 544 x 544 input images.

The author highlighted that higher resolution images for the same model have better mAP but slower time processing.
Despite the shifting scenario and the different optimization techniques applied, its possible to notice:

- Region based detectors (like Faster R-CNN) demonstrate a small accuracy advantage if real-time speed is not needed;
- Single shot detectors work well for real-time processing. However, it's necessary to verify whether its accuracy meets the respective accuracy requirement.

The research made in [HRS+17] offers a survey to study the tradeoff between speed and accuracy for:

- Faster R-CNN;
- R-FCN;
- SSD.

It re-implements those models in TensorFlow using Microsoft COCO dataset for training, and also introduces MobileNet, which achieves high accuracy with lower complexity.
Figure 2.14 shows the results obtained the overall mAP and the respective GPU time for each configuration tested.

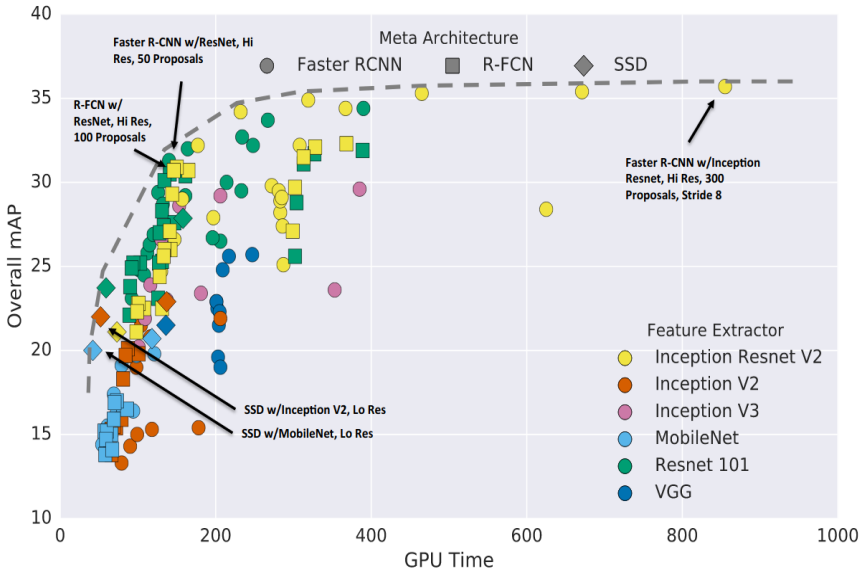


Figure 2.14: Accuracy vs time, with marker shapes indicating meta-architecture and colors indicating feature extractor. (Source: [HRS+17]).

The most accurate was Faster R-CNN with Inception ResNet V2 and 300 proposals. Considering the time, we can see that SSD with MobileNet and SSD with Inception V2 (both with low resolution) were the fastest models. Also, it was possible to see sweet spots, for example, Faster R-CNN with ResNet-101 and 100 proposals, and R-FCN with ResNet-101 and 300 proposals.

Figure 2.15 shows the behavior of object detector accuracy based on the feature extractor accuracy. To avoid crowding the plot, the authors showed only the low resolution models.

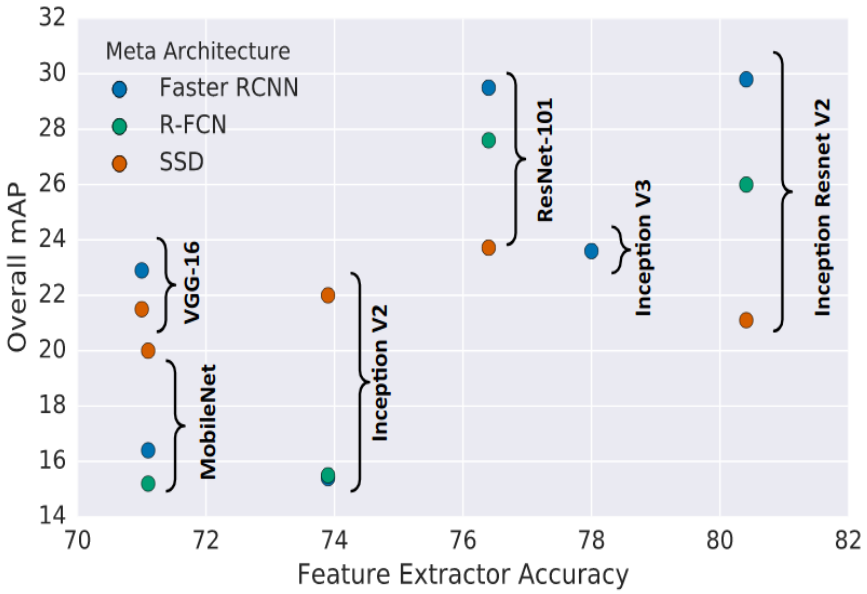


Figure 2.15: Accuracy of detector (mAP on COCO) vs accuracy of feature extractor (as measured by top-1 accuracy on ImageNet-CLS). (Source: [HRS+17]).

This experiment aims to study the influence of the feature extractor's accuracy on the detector's accuracy. From the results it was possible to see that Inception ResNet V2 has less impact with SSD (21/81)[1]. However, with R-FCN and Faster R-CNN, it's possible to get better results: (26/81), and (30/81), respectively.

Overall, VGG-16 presented the lowest values (with all object detectors). SSD and R-FCN obtained their best mAP values with ResNet-101: 24 and 28, respectively.

Figure 2.16 shows the behavior of object detector accuracy based on the object size. The authors fixed the image resolution to 300.

It's possible to see that all detectors perform very well with large objects. However, SSD presented a very poor performance with small objects, with highest value of approximately 3%. Figure 2.17 shows the behavior of each object detector.

It's possible to see that the SSD has problems in detecting small objects (for example: bottles, cup, bag) in the middle of the table, while other methods do not.

Figure 2.18 shows the object detector accuracy based on the object size. The authors studied the resolutions 300 and 600.

The authors concluded that the input resolution can significantly impact detection accuracy. They observed that decreasing resolution by a factor of two in both dimensions results in lowers accuracy (15.88% on average), but also reduces inference time (27.4% on average).

Figure 2.19 shows the behavior of object detector accuracy based on the number of proposals.

It's possible to see that for Faster R-CNN with Inception ResNet, they obtained 96% of the

[1](Object detector accuracy/feature extractor accuracy)

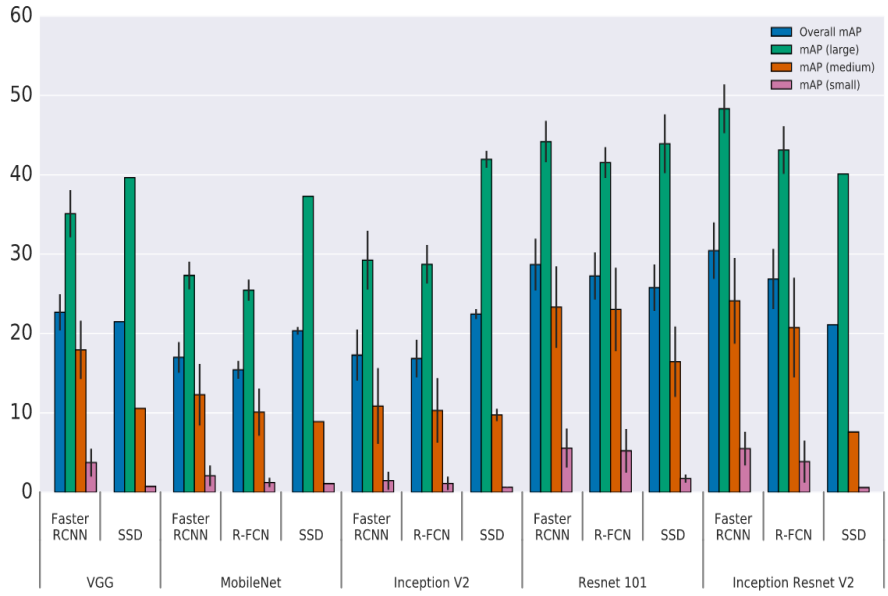


Figure 2.16: Accuracy stratified by object size, meta-architecture and feature extractor. (Source: [HRS+17]).

(a) SSD+Mobilenet, lowres

(b) SSD+InceptionV2, lowres

(c) FRCNN+Resnet101, 100 proposals

(d) RFCN+Resnet10, 300 proposals

Figure 2.17: Example from 4 different modelsce: [HRS+17]).

accuracy of using 300 proposals. However, using only 50 proposals is possible to improve the speed by a factor of 3, and the accuracy drops only 4%. This is because R-FCN has much less work per ROI, hence the speed improvement is far less significant.

The authors also presented the GPU time for each model combination. However, due to the platform dependency, they decided to count FLOPS, so it gives a platform independence measurement of computation, which may or may not be linear concerning actual running times due to several issues such as caching, input/output, hardware optimization etc.

Figure 2.20 shows the FLOPS count against observed wall-clock times on the GPU and Central Process Unit (CPU), respectively.

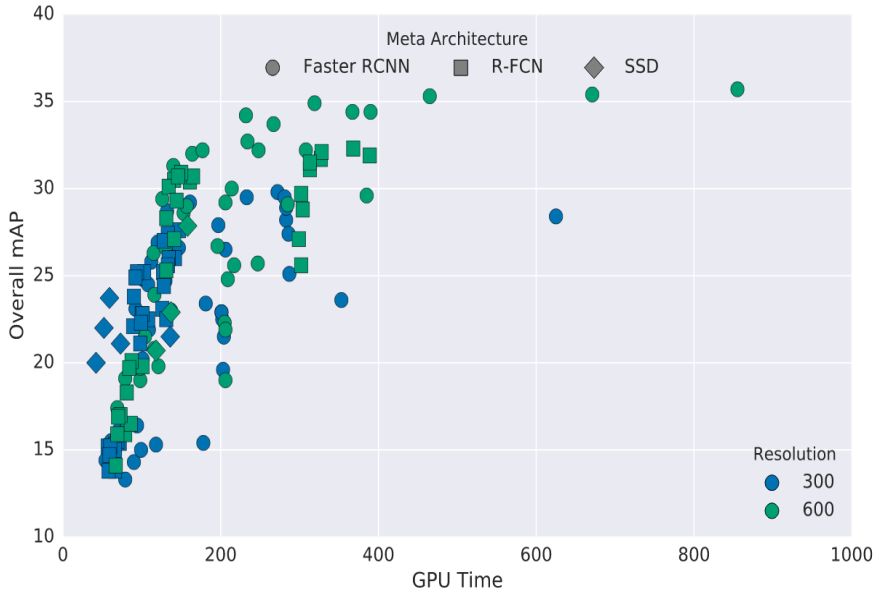


Figure 2.18: Result of the effect of image resolution. (Source: [HRS+17]).

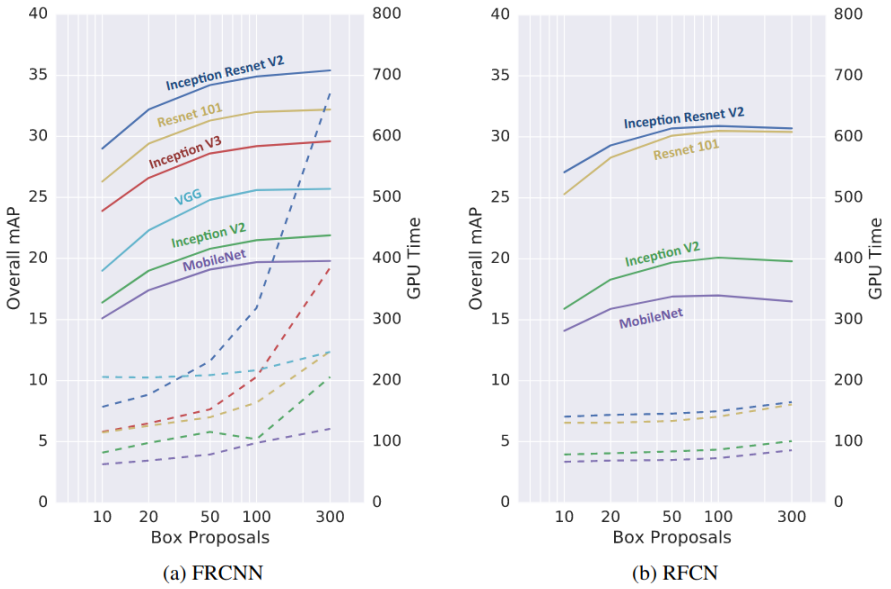
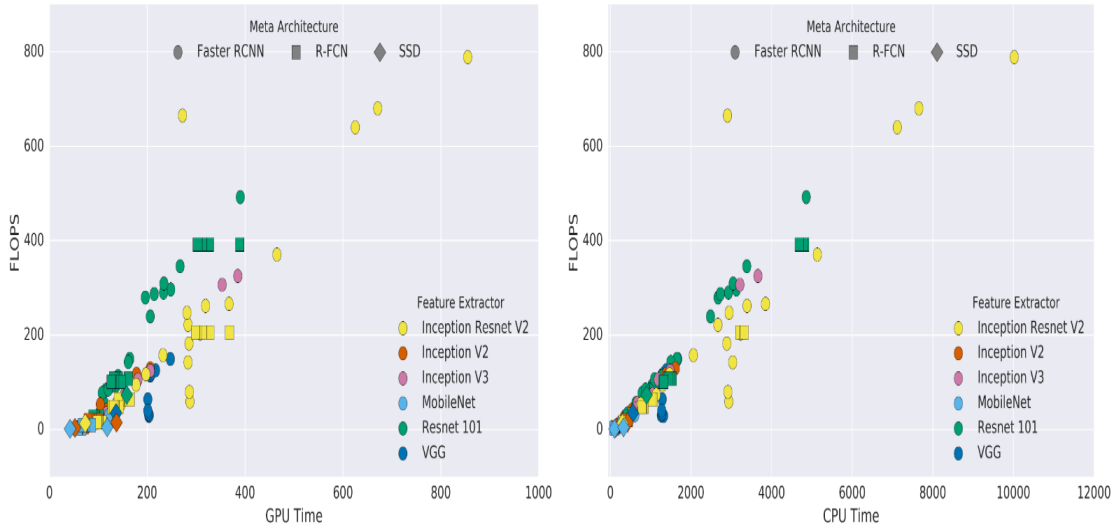


Figure 2.19: Effect of proposing increasing number of regions on mAP accuracy (solid lines) and GPU inference time (dotted). (Source: [HRS+17]).

The authors measured the total memory usage, also they included all datapoints corresponding to the low-resolution models. The error bars reflect variance in memory usage by using different numbers of proposals for the Faster R-CNN and R-FCN models (which leads to the seemingly considerable variance in the Faster R-CNN with Inception ResNet bar).

Figure 2.21 shows the memory usage for each model. It's possible to see that MobileNet requires the lowest amount of memory. Otherwise, Faster R-CNN with Inception ResNet V2 presented the highest use of memory.

From this study, it was possible to highlight some points:



(a) GPU. (b) CPU.

Figure 2.20: FLOPS vs GPU time. (Source: [HRS+17]).

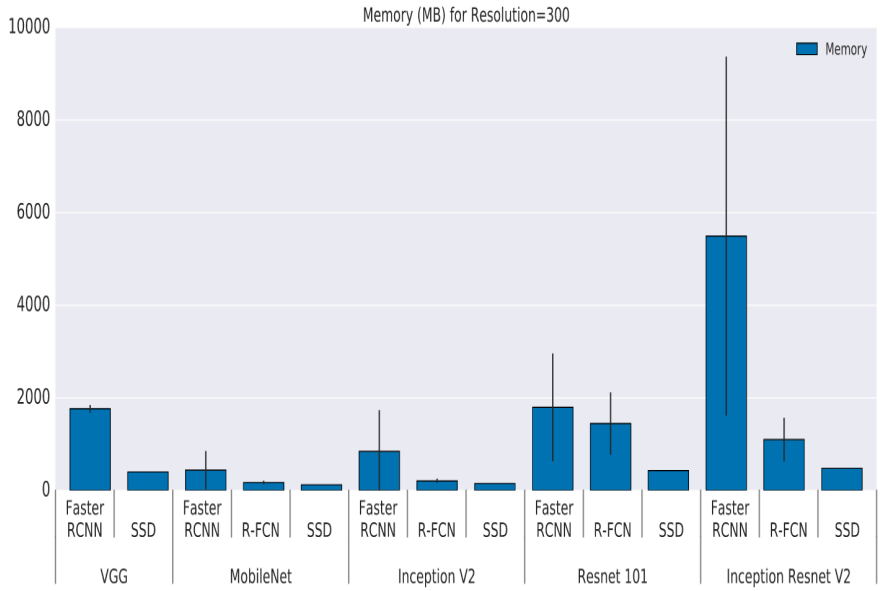


Figure 2.21: Memory (Mb) usage for each model. (Source: [HRS+17]).

- R-FCN and SSD models are faster. However, the Faster R-CNN is better in terms of accuracy if speed is not a problem;
- Faster R-CNN requires at least 100 ms per image;
- Reduce image size by half in width and height lowers accuracy by 15.88% on average but also reduces inference time by 27.4% on average;
- Choice of feature extractors impacts detection accuracy for Faster R-CNN and R-FCN but less reliant for SSD;
- The most accurate single model use Faster R-CNN using Inception ResNet with 300 proposals;

- SSD with MobileNet provides the best accuracy tradeoff within the fastest detectors;
- SSD is fast but performs worse for small objects comparing with others;
- For large objects, SSD can outperform Faster R-CNN and R-FCN in accuracy with lighter and faster extractors;
- Faster R-CNN can match the speed of R-FCN and SSD at 32mAP if we reduce the number of proposal to 50.

Thus, using object detectors we can not only detect salient regions but also, through the detected classes, we can get a clue about the image's context. Figure 2.22 shows an example where the use of an object detector could alert the authorities. In March 2019, two young men broke into a school in Suzano, Brazil [Var20]. They commited a massacre, which left five students dead. Days before the massacre took place, one of the shooters posted photos with the gun that would be used in the attack [Qui19].

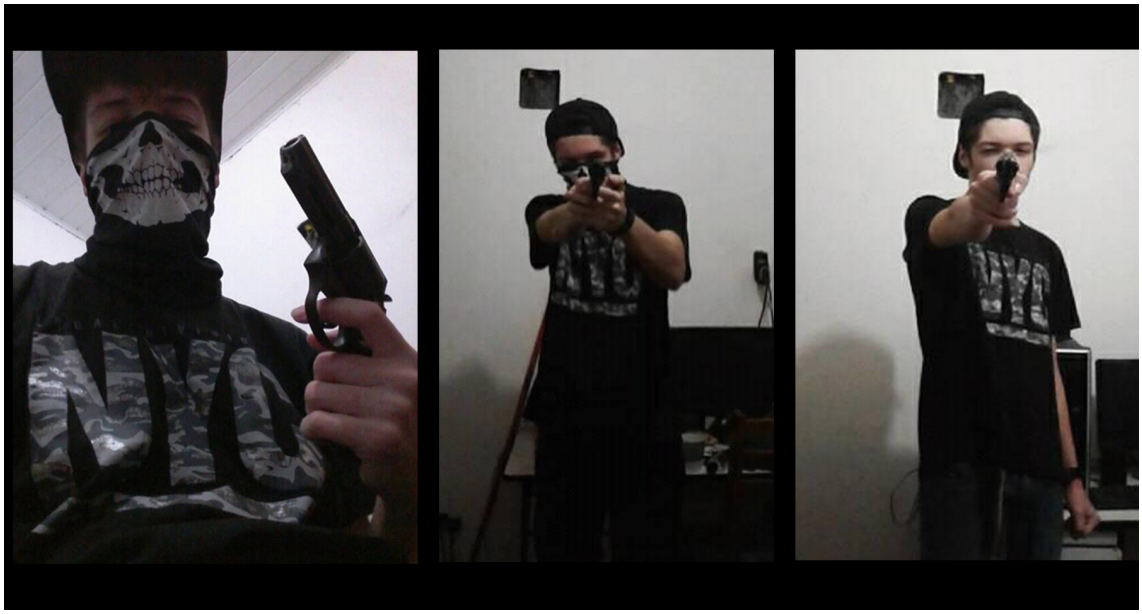


Figure 2.22: The shooter responsible for the Suzano school's massacre, posted photos with the gun before the crime. (Source: image from [Dia19]).

2.5 Facial Expressions Classification Algorithms

FER, as the primary processing method for non-verbal intentions, is an important and promising field of computer vision and artificial intelligence [HCLW19]. Facial expression recognition is the task of classifying the expressions on face images into various categories such as anger, fear, surprise, sadness, happiness, and so on. Several works affirm that bright images, normally, are positive. However, this isn't always true. Sometimes the global information can trick the model, and it leads to a wrong classification. Figure 2.23 gives us an example, with the global information of the image, the model might predict it

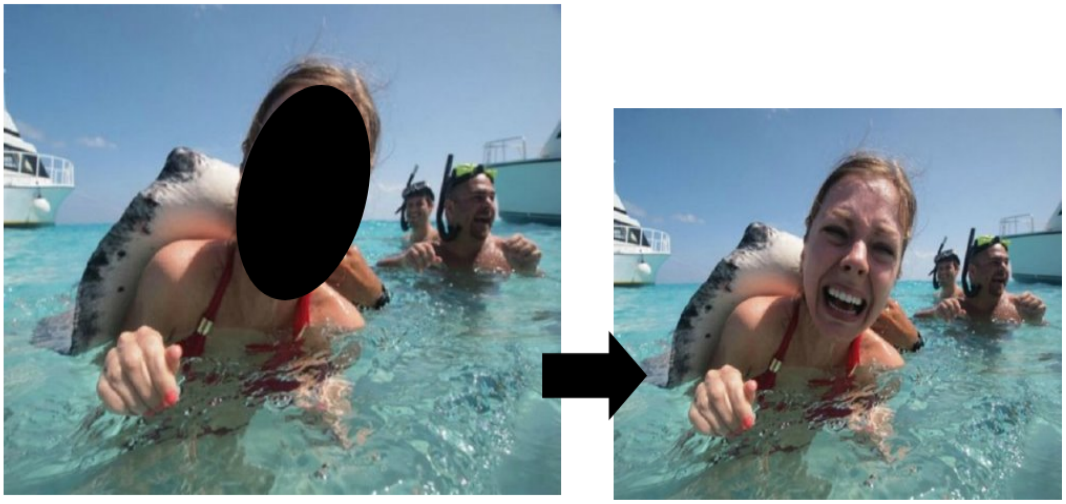


Figure 2.23: Illustration which represent the features' influence in prediction, and how the facial expression analysis can help the model's performance. (Source: image from [des18]).

as positive. However, looking at the detail of the facial expression, we can clearly see that the woman is desperate.

To get familiarized with the facial expression recognition methods, a brief research was made, and it will be presented.

In [HCLW19], an overview of recent advances in FER is presented. The most popular models are AUs, which encode basic movements of facial muscles, and V-A space, which identify emotion categories according to the value of the emotion dimensions (reminding the circumplex model presented in subsection 2.6.3). The authors based their discussion on the AUs model. Figure 2.24 shows an example of the AUs.

Figure 2.24: Some examples of AUs. (Source: image from [HCLW19]).

The study about FER can be divided into two groups:

- **Conventional FER approach:** is composed of three major steps - image pre-processing; feature extraction, and expression classification;
- **Deep learning-based FER approach:** reduces the dependence feature extraction by employing an "end-to-end" learning directly from input data to classification result.

When compared to the methods presented in the subsection 2.3, the methods based on manual feature extraction are less dependent on data and hardware, which have advantages in small data sample analysis. While the deep learning approaches demand less manual work, they need massive datasets to avoid overfitting. The data can, in general, be divided into two groups: laboratory type, and wild type.

2.5.0.1 Conventional FER Approaches

As briefly seen in subsection 2.3, a notable characteristic of the conventional FER approach is its high dependence on manual feature extraction. Thus, the procedure in conventional FER approaches can be divided into three tasks:

- **Image pre-processing:** aims to eliminate irrelevant information from the input images and improve the detection ability of relevant information. This step has a high influence on the extraction of features and the performance of expression classification. For this step, the authors presented four sub-tasks to be performed: i) noise reduction; ii) face detection; iii) normalization; iv) histogram equalization;
- **Feature extraction:** aims to extract useful information from the image. Some of the main methods used for feature extraction in FER are: Gabor feature extraction, Local Binary Pattern (LBP), optical flow method, Haar-like feature extraction. As seen in the section 2.7, and subsection 2.4, the feature extraction influences the performance of the algorithm;
- **Expression classification:** aims to select the appropriate classifier that can predict the face expressions. Some of the main applied classifiers in FER are: k-Nearest Neighbours (kNN), SVM, Adaptive Boosting (Adaboost), Sparse Representation-based Classifier (SRC), Bayesian, and Probabilistic Neural Network (PNN).

Despite the dependency on data and hardware, the feature extraction and classification have to be designed manually and separately. Thus, the effectiveness of conventional FER methods depends on the performance of each component.

2.5.0.2 Deep Learning-Based FER Approaches

Deep learning has demonstrated outstanding performance in many machine learning tasks including identification, classification, and target detection [HCLW19]. According to the

authors, this approach deeply reduces the dependence on image pre-processing and feature extraction. Also, they are more robust to the environment with different elements (illumination and occlusion). Some of the main approaches are:

- **CNN:** is an "end-to-end" model, an improvement of the Artificial Neural Network (ANN) [WRP+17]. The qualities of CNN include local connectivity and weight sharing, resulting in faster training speed, and regularization effect;
- **Deep Belief Network (DBN):** is based on Restricted Boltzmann Machine (RBM) and its feature extraction of the input signal is unsupervised and abstract. The methods based on DBN can learn the abstract information of facial images automatically and are sensitive to activity factors;
- **LSTM:** the methods based on LSTM are well suited for temporal feature extraction of sequential frames. This can be a good choice to be used on video sequence analysis;
- **Generative Adversarial Network (GAN):** is an unsupervised learning model composed of a generative network and a discriminative network. The models based on GAN not only contribute to training data augmentation and the recognition tasks, but also for pose-invariant and identity-invariant expression recognition.

The authors presented some FER-related datasets. However, we are more interested in the images of the dataset, so any video or frame information will not be presented. Table 2.7 shows an overview of the related dataset in the FER area.
The authors did a comparison of representative FER approaches on widely evaluated datasets. In the comparison, all the approaches and respective accuracy were presented. However, we will only present the approach that achieved the highest accuracy. Table 2.8 shows the approaches that obtained the highest accuracy on the widely evaluated datasets.
Concluding the survey, some challenges found in FER (theoretical and practical) were presented, for instance:

- **Wild Environmental Conditions:** complex conditions like occlusion and pose-variation, which may delay the recognition of original facial expressions, are two major obstacles to the versatility of FER, especially in wild scenarios;
- **The Lack of High-Quality Publicly Available Data:** mainly with deep learning-based approaches, it usually requires a large amount of training data to capture subtle expression-related deformations. The major challenge these approaches suffer is the deficit of training data in terms of both quantity and quality;
- **Visual Privacy:** increasing privacy-preserving concerns are a major obstacle in camera-equipped systems. Consequently, more reliable and accurate privacy protection methods are required to discover a balance between privacy and data utility for FER models.

Dataset	Year	Subjects	Material	Resolution	Condition	Elicitation	Annotation
Japanese Female Facial Expressions (JAFFE)[LAKG98]	1998	10	213 still images	256 x 256	Unique	Posed	6 Basic Emotions (BEs) & Neutral
Extended Cohn–Kanade (CK+)[LCK+10]	2010	123	593 still images	640 x 480	Unique	Posed	6 BEs & contempt
Compound Emotion (CE)[DM15]	2015	230	5,060 still images	3000 x 4000	Unique	Posed	22 CEs
Maja & Michel Initiative (MMI)[PVRM05]	2005	75	740 still images	720 x 576	Complex	Posed	6 BEs & Neutral, AUs
Binghamton University 3D Facial Expression (BU-3DFE)[YWS+06]	2003	100	2,500 still images	1040 x 1329	Complex	Posed	6 BEs, 83 Facial Landmarks (FLs)
MPI[KCBW12]	2012	19	1,045 still images	768 x 576	Complex	Spontaneous	55 expressions
Karolinska Directed Emotional Face (KDEF)[LF□98]	1998	70	4,900 still images	562 x 762	Complex	Posed	6 BEs & Neutral
Multi-Pose, Illumination, and Expression (Multi-PIE)[GMC+08]	2013	337	755,370 still images	-	Complex	Posed	6 expressions, FLs
Oulu-CASIA[ZHT+11]	2011	80	2,880 image sequence	320 x 240	Complex	Posed	6 BEs
FER2013[GEC+13]	2013	-	35,887 still images	48 x 48	Wild	Posed & spontaneous	18 expressions, 6 BEs & Neutral
Static Facial Expressions in the Wild (SFEW)[DGLG11]	2011	-	1,766 still images	-	Wild	Posed & spontaneous	6 BEs & Neutral
Real-world Affective Database (RAF-DB)[LDD17]	2017	-	29,672 still images	-	Wild	Posed & spontaneous	6 BEs & Neutral, 12 CEs, 42 FLs
Real-world Affective Faces Multi Label (RAF-ML)[LD19]	2018	-	4,908 still images	-	Wild	Posed & spontaneous	6 BEs & Neutral, 12 CEs, 42 FLs
GENKI-4K[WLFM09]	2009	-	4,000 still images	-	Wild	Spontaneous	6 BEs distribution vector, 42 FLs (smiling & non-smiling), head-pose

Table 2.7: An overview of FER-related datasets, based on the survey [HCLW19].

Dataset	Approach	Accuracy
JAFFE[LAKG98]	SVM[TC18]	97.10%
CK+[LCK+10]	CNN[BK17]	98.62%
MMI[PVRM05]	3D Inception-ResNet (3DIR) + LSTM[HM17]	79.26%
FER2013[GEC+13]	CNN[BK17]	72.10%
BU-3DFE[YWS+06]	GAN[YCY18]	84.17%
Multi-PIE[GMC+08]	Deeper CNN[MCM16]	94.70%
SFEW[DGLG11]	Anatomically Constrained Neural Networks (ACNN)[LZSC19]	51.72%
Oulu-CASIA[ZHT+11]	GAN + CNN[YZY18]	88.92%

Table 2.8: Comparison of representative FER approaches on widely evaluated datasets, based on the survey [HCLW19].

2.6 Sentiment Models

The lack of proper differentiation between affect, feeling, emotion, sentiment, and opinion, and understand how they relate to one another. The work [MSMSP14] aims to clarify the difference between these terms and present significant concepts to the computational linguistics community for their effective detection and processing. Affect, feeling, emotion, sentiment, and opinion are terms relating to human subjectivity, which is a feature of the person's mind. Thus, like in the image sentiment analysis, these subjective experiences will be created from the subject perspective (and we need to take into account the cultural background, societal background, and other factors that may influence this perspective), and they will reflect the subject's desires, beliefs, and feelings. The authors complain about the lack of consistency in terminology, which means the fact that these subjective terms are used without sufficient differentiation between them, and this leads to a poor apprehension and confusion about what concepts should be involved in text analysis. Also, they complain about works in Sentiment Analysis (SA) and Opinion Mining (OM) focused on detecting text polarity, arguing that sentiment and opinions are more complex than just having polarity.

The authors presented the definitions found in dictionaries of these subjective terms. Table 2.9 presents the definitions presented by the authors in [MSMSP14], which was based in the Merriam-Webster Online Dictionary [mer28] accessed in 2014 (to avoid any outdated definition, the dictionary was accessed to compare the definition presented in 2014 with the definitions found nowadays).

It's worth mentioning that the definitions presented are related to the subjective term, for example, the feeling can be defined as the basic physical sense (touch), but we are not interested in definitions like this. There is no information if the definitions suffering updates or if they passed through a review, but for Affect (noun), none of the synonyms found were 'Feeling', moreover 'Feeling' was presented as obsolete, which is a term used to the definitions that will be encountered when visiting the literature of the past [obs].

Subjectivity Term	Definition	Synonym
Affect	The conscious subjective aspect of an emotion considered apart from bodily changes; also a set of observable manifestations of a subjectively experienced emotion.	Feeling*
Feeling	An emotion state or reaction; often unreasoning opinion or belief.	Sentiment, Emotion
Emotion	Excitement; the affective aspect of consciousness; a state of feeling; a conscious mental reaction (as anger or fear) subjectively experienced as strong feeling usually directed toward a specific object and typically accompanied by physiological and behavioral changes in the body.	Feeling, Sentiment
Sentiment	An attitude thought, or judgment prompted by feeling; a specific view or notion.	Feeling, Emotion, Opinion
Opinion	A view, judgment, or appraisal formed in the mind about a particular matter; A belief stronger than impression and less strong than positive knowledge.	Feeling, Sentiment

Table 2.9: Definitions provided by Merriam-Webster Online [mer28].

However, is possible to see, through the synonyms, where the confusion begins.

The authors presented a scheme to show the differentiating factors of subjectivity terms. Figure 2.25 shows the respective scheme.

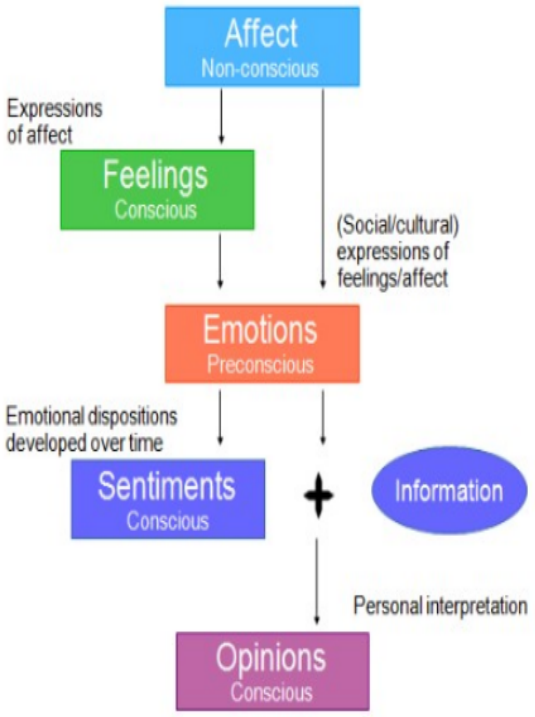


Figure 2.25: Differentiating factors between the subjective terms. (Source: image from [MSMSP14]).

The work [KK18] presents an overview of the existing body of research on sentiment and emotion analysis. However, the focus will be on popular models and theories in computational analysis.

2.6.1 Ekman's Theory of Basic Emotions

Ekman, Soreson, and Friesen (1969) proclaimed that facial displays of fundamental emotions are not learned but innate. However, there is the cultural influence that leads to how and which situations emotions are displayed (the facial expression that best fits). Also, Ekman's theory postulates that emotions should be considered as discrete categories rather than continuous. Thus, Ekman devised his list of basic emotions after researching many different cultures, and found 6 basic emotions, [Han]: anger, disgust, fear, happiness, sadness, and surprise.
In the 1990s he added other emotions but stated that not all of these can be encoded via facial expressions: amusement, contempt, contentment, embarrassment, excitement, guilt, pride in achievement, relief, satisfaction, sensory pleasure, and shame.

2.6.2 Plutchik's Wheel of Emotions

In the early 1980s, Robert Plutchik proposed a model of emotions, which became popular. Unlike Ekman's theory, Plutchik's theory defended a small set of basic emotions, and all other emotions are the result of the mix and derived from the various combinations of basic ones. Plutchik's Wheel of Emotions has the following characteristics:

- Similar emotions are placed closer together;
- Opposite emotions are placed 180 degrees apart;
- The intensity of an emotion depends on how far from the center a part of a petal is.

The wheel is constructed from eight basic bipolar emotions, as shown in Figure 2.26:

- Joy vs sorrow;
- Anger vs fear;
- Trust vs disgust;
- Surprise vs anticipation.

The blank spaces between the leaves are emotions that are mixture of two of the primary emotions:

- **Optimism** = joy + anticipation;
- **Aggressiveness** = anticipation + anger;
- **Contempt** = anger + disgust;
- **Remorse** = disgust + sadness;
- **Disapproval** = distraction + pensiveness;
- **Awe** = fear + surprise;
- **Submission** = fear + trust;
- **Love** = trust + joy.

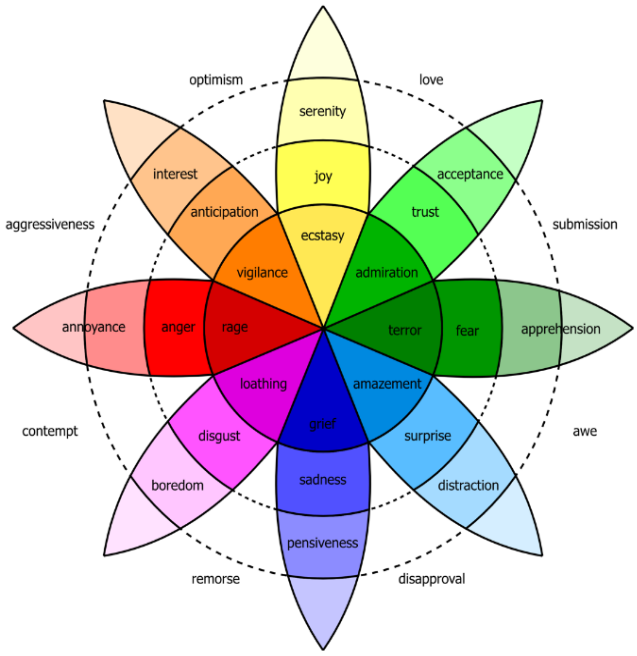


Figure 2.26: Plutchik's wheel of emotions. (Source: image from [KK18]).

2.6.3 Russel's Circumplex Model

James Russel proposed a circumplex model of affect aiming to overcome the shortcomings of basic emotions theory. The "circumplex" is related to the fact that emotional episodes do not cluster at the axes but rather at the periphery of a circle. At the core of the circumplex model is the notion of two dimensions plotted on a circle along horizontal and vertical axes. These dimensions are:

- **Valence:** how pleasant or unpleasant one feels;
- **Arousal:** the degree of calmness or excitement.

The number of dimensions is not strictly fixed. Each affective experience can be depicted as a point in a circumplex that is described by two parameters (valence and arousal) without the need for labeling or reference to emotion concepts for which a name might only exist in particular subcommunities, or which are difficult to describe [Rus03]. Figure 2.27 shows the circumplex model of affect proposed by Russell.
However, it is not clear what should be done with qualitatively different events, for example, fear and disgust will fall in identical places in the circumplex structure [RB99]. Thus, this model is applied when the interest is in continuous measurements of valence and arousal rather than in the specific discrete emotional categories.

2.6.4 Parrots' Classification of Emotion

In 2001 a tree-structured list was proposed by Parrot, where over 100+ emotions were identified and conceptualized [Han] the first level is composed of six primary emotions. This classification differs from the others above cited because the secondary emotions are

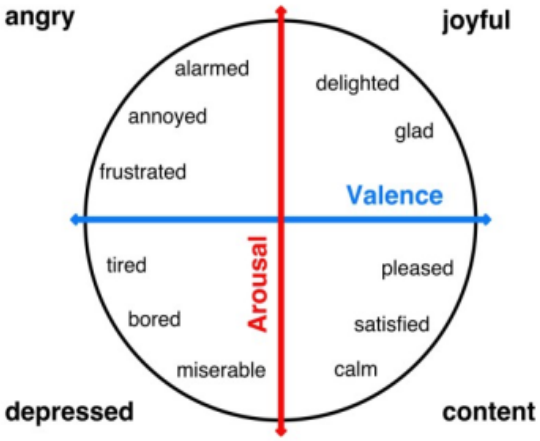


Figure 2.27: Circumplex model of affect. (Source: image from [KK18]).

the derivation of the primary ones instead of being a combination of them. Figure 2.28 shows the first two layers of Parrot's classification.

Figure 2.28: First two layers of the Parrots' Classification. (Source: image from [BDLM13]).

2.6.5 Hugo Lövheim Cube of Emotions

Hugo Lövheim proposed in 2011 a classification that merger the categorical and the dimensional: the cube representation. In this representation, eight basic emotions are ordered in an orthogonal coordinate system of three main monoaminergic axes, which represent serotonin, dopamine, and noradrenaline [BDLM13]. These neurotransmitters compose part of the monoamine class [BDLM13], [Pim]. Figure 2.29 shows its 3D repre-

sentation.

- **Noradrenaline** is a neurotransmitter that plays an important role in responses that involve fight or flee, your production increases with danger or stress situations;
- **Dopamine** plays an important role in the coordination of body movements, also is involved in reward, motivation, and reinforcement;
- **Serotonin** plays an important role in self-confidence, inner strength, and satisfaction.

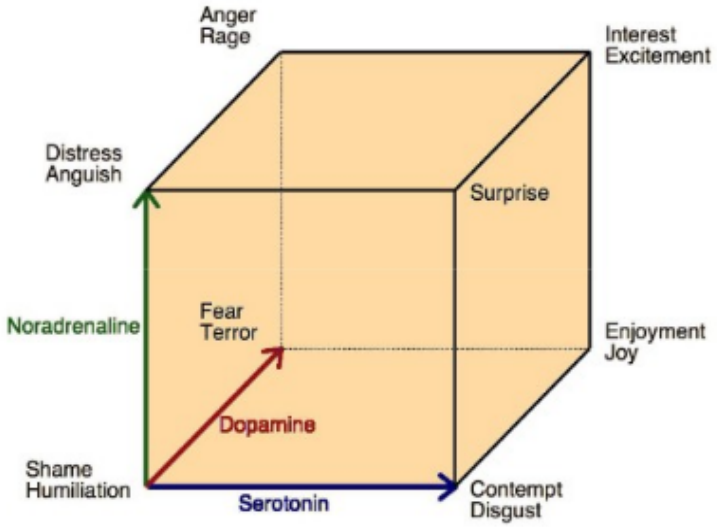


Figure 2.29: The cube of emotions proposed by Hugo Lövheim. (Source: image from [Pim]).

2.7 Related Work

The work [YLJY15b] tries to handle the image sentiment analysis (without any text consideration), using a Plutchik's wheel of emotions approach to classify those emotions. Also, address challenge issues like implementing supervised learning with weakly labeled training data, in other words, data that was labeled through a model (and not labeled by a human), and handles the image sentiment classification generalizability. Thus, the work focused on finding the resolution for two main questions:

- Implement the supervised learning (because of the CNN) with weakly labeled training data;
- Reach a high level of generalizability to cover different domains (we can understand domains as the emotions).

The progression's implementation aims to reduce the impact of training instances' noise, and to allow the model to have a high level of generalizability. Firstly, the CNN will be

trained with Flickr's images, then a subset will be selected from the difference between the prediction's values and training data itself, in that way, the model will be fine-tuned. The smaller the difference, the higher the probability of the instance to be removed from the training set. In the end, this fine-tuned model will be the final model to make the image sentiment analysis. The experiments aim to solidify the use of progressive CNN by comparing it with different CNN architectures (networks with iCONV-jFC). Also, it was compared the Progressive Convolutional Neural Network (PCNN)'s performance with the other three baselines or competing algorithms for image sentiment classification. The results are evaluated by the following metrics:

- **Precision:** the number of the correct results divided by the number of all returned results – how many of them are actually positive;
- **Recall:** the number of the correct results divided by the number of results that should have been returned;
- **F1:** used when we want to seek a balance between Precision and Recall. It is the harmonic mean of Precision and Recall and gives a better measure of the incorrectly classified cases;
- **Accuracy:** it is the measure of all correctly identified cases. It is mostly used when all the classes are equally important.

From the tests, it was confirmed that the fine-tune applied on CNN allowed the network to find out a better local optimum and improves the generalizability extensibility. Even PCNN presenting the best result in all tests, there is a deficit in not presenting the computational cost and the execution/training time to each model. Also, it would be interesting to test other architectures approaches, for instance: bottom-up, lateral, and top-down connections, [SMK17]. The work concludes that the use of CNN provides advantages over using predefined low-level visual features or mid-level visual attributes. The main advantage presented by using CNN is the possibility to transfer knowledge to other domains by using the fine-tune strategy. Another significant point is the possibility of using weakly labeled data in the training set.
The work [GA19b] aims to reduce the image classification's dependence on the text content. The proposed model was divided into 3 parts (in each one there is a specific task) and then, in the end, all parts are fused using a weighted sum, which is capable of predicting the polarity of a sentiment level (positive, neutral, and negative). Those three parts are:

- **Text analysis:** two methods were tested, Vader and TextBlob, using the respective confusion matrices, and considering the results that the B-T4SA dataset provides on a validation set, the authors in [GA19b] concluded that TextBlob reveals a higher accuracy.
- **Image analysis:** an exploration of ResNet topology was made, in which three versions were selected (ResNet18, ResNet50, and ResNet152), this choice was justified by arguing that this method reduces significantly the vanishing gradient problem.

From a test, the ResNet152 presented the best performance, which outperformed the results presented in another work [VCC+17a]. To use this model, the dataset (with many images) was prepared (re-sizing each image to 224 x 224) and all hyper-parameters were set up.

- **Content image analysis:** a pre-trained (InceptionResNetV2) model with the ImageNet was used to classify the data. The content image analysis aims to build a probability distribution that allows them to classify an image according to its sentiment polarity.

In the end, a weighted sum is calculated, and the information fusion is made through a voting system, where each method (above cited) has specific importance in the vote.
To execute the experiments, the authors of [GA19b] chose to use a subset from the T4SA dataset, called B-T4SA, which is divided into three partitions (train, validation, and test) where each class has the same number of images. With the proposed method, the authors obtained 52.20% accuracy on the test set, which outperformed the results presented in [VCC+17a].
The work [GA19a] aims to evaluate image classification methods to improve the state of the art in a large tweet dataset by using different approaches to improve the results obtained from its previous work [GA19b], focusing only on the analysis of sentiment on isolated images. In the previous work, the authors explored three versions of the ResNet, which are: the ResNet18, the ResNet50, and the ResNet152. Now they added the other two architectures: InceptionV3 and DenseNet. As well as in the previous work, the data needed to pass through a re-sizing process, thus, the pre-processing method resized each image to 224 x 224 (ResNet architectures) and 299 x 299 in the others. Also, all the model was set up with the same hyper-parameters, which have the same values from the previous work, and the B-T4SA was used. About the new results (obtained with Inception V3 and DenseNet), it was concluded that is possible to improve the accuracy value with DenseNet, with an increase of 0.76%. Table 2.10 presents the results obtained.

		Network Architectures				
		ResNet18	ResNet50	ResNet152	InceptionV3	DENSENet61
Train	Time(H)	**6**	17	41	25	78
Test	LOSS	1.0490	0.9909	0.9821	0.9770	**0.9730**
	ACC	0.4474	0.5251	0.5234	0.5156	**0.5274**

Table 2.10: Comparison between the results obtained from the tests.

It's possible to see the accuracy improvement. However, looking at the time, it's possible to conclude that ResNet50 reaches a similar value, but in a shorter time. That's interesting because, in the previous work, ResNet152 presented the best result (52.20%). The authors concluded that it is possible to improve the accuracy, however, the time question must be considered depending on the application.

This work [ZWS+20] proposes a novel model for image sentiment analysis. The authors identified two key issues that need to be addressed, which are: i) high-quality training samples are scarce; ii) the cross-modal sentimental semantic among heterogeneous image features have not been fully explored. To reach this goal, the authors proposed a novel model called Multidimensional Extra Evidence Mining (ME^{2}M), which contains three main pieces of evidence:

1. New sample-refinement strategy to refine the "Pending Data" (data waiting for decisions or refinement);
2. Extraction of a set of complementary image features including traditional image features and deep-learning based features;
3. The use of the Different of Convex functions Algorithm (DCA) model to complete cross-modal sentimental semantics mining.

The use of cross-modal sentimental semantics is encouraged because according to the authors:

- There is the possibility to perform image sentiment analysis through heterogeneous image features;
- Can depict the key visual contents of images accurately and comprehensively.

The authors used 2 datasets to execute the experiments, which are: Twitter I and FI. With the experiments, the authors concluded that:

1. **MEM model with cross-modal sentimental semantic:**
    - The shape (S) feature is the most important visual cue for characterizing the Twitter I dataset, and SV19 delivered the highest average accuracy value, and XGBoost20 presented the highest average accuracy between the classifiers;
    - The deep-learning based feature (V) is the most important visual cue for characterizing the FI dataset, and V19V16 delivered the highest average accuracy value, and Logistic Regression presented the highest average accuracy between the classifiers.
2. **ME^{2}M model with cross-modal sentimental semantic and sample-refinement:**
    - SV19 delivered the highest average accuracy value, and Naive Bayes presented the highest average accuracy between the classifiers;
    - V19V16 delivered the highest average accuracy value, and Logistic Regression presented the highest average accuracy between the classifiers;
    - The authors also compared the ME^{2}M model with the first category baseline. For the Twitter I dataset, the ME^{2}MKNN model achieves the highest performance improvement. For the FI dataset, the ME^{2}MLR model achieves the highest performance improvement.

3. **Classification performance comparisons between the ME^{2}M model and several state-of-the-art baselines:**
    - The DCA algorithm outperforms other popular cross-modal analysis model and can promote the real-time efficiency of the ME^{2}M model;
    - The idea of cross-modal sentimental semantics mining is more important than the proposed sample-refinement strategy on a relatively small dataset;
    - The proposed sample-refinement strategy plays a more important role in a relatively large dataset.
4. **Qualitative analysis:**
    - The concatenation mode can maximally retain the key discriminant information of transformed features;
    - Deep learning-based features are insufficient for characterizing sentimental semantics.

To predict the image sentiments, the authors in [WQJZ20b] proposed a model that combines global and local information. The work proposes a framework to leverage local regions and global information to estimate the sentiment conveyed by images, which the same pre-trained CNN model will be used, but it will be fine-tuned using different training sets. Training set I will address the entire images (GM_{EI}), and training set II will address the sub-images (LRM_{SI}), in the end, both predictions will be fused to obtain the final sentiment prediction. The extraction of the sub-images is based on the detection window for salient objects in the entire image.
To evaluate the performance the authors conducted the experiments on five datasets. The first experiment was designed to obtain α's optimal value (and 0.8 was obtained). To employ binary classification the authors mapped into two labels the multi-emotion labels. To evaluate the use of local information, the authors compared two schemes: i) a model trained on entire images – GM_{EI}; ii) two models trained on entire images and sub-images, respectively – GM_{EI}&LRM_{SI}. This latter scheme presented always the higher value accuracy to all datasets.
To evaluate the sentiment classification performance, the authors compared their model with 4 state-of-art algorithms, which are: PCNN [YLJY15b], VGGNet (GM) [YSS$^+$18], DeppSentiBank [CBDC14], and Yang's method [YSS$^+$18]. All these algorithms use only global information. With the results, the authors concluded that their model outperforms the other algorithms.
Also, the authors evaluated if local information is always effective on sentiment prediction. They analyzed 140 images, of the EmotionROI dataset, in the testing set (total 590 images) that don't include any salient object and its performance with/without the local information. To extract a sub-image, without having salient objects in the image, they are based on normalized Emotion Stimuli Map in EmotionROI. Using only global information, the model reached 75.71% of accuracy. However, using only local information, the accuracy drops to 73.57%.

The authors in [FCd19] propose a method based on a multitask framework to combine multimodal information whenever it's available, the explanation for their choice it's because in real environments (social media) we seldom have both image and text information available simultaneously. Thus, they want to propose a model that is able to handle the cases where a modality is missing.
For this problem, the authors commented on some solutions, which are:

- **The late-fusion.** However, this approach suffers due to its simplicity and address the modalities independently and can't learn discriminative multimodal interactions;
- **A robust method to a missing modality.** However, this approach requires a complex training strategy.

The proposed model contains one classifier for each task:

- Text classification;
- Image classification;
- Prediction based on the fusion of both modalities.

The authors explain this approach can overcome the problems of the previous approaches cited, because:

1. The multimodal classifier can use any fusion technique to learn complex multimodal interactions;
2. The monomodal classifiers enable the model to perform accurate predictions (even with there is no text-image pair available).

The performance could be improved because the feature extractors and the monomodal classifiers to be trained with image-only or text-only examples.
The model must be able to predict the label y^i of an instance x^i, where label is the sentiment or an emotion. Also, the model should be able to handle with an instance x^i (at training and test time) even if it is an image, a text or an image-text pair.
Figure 2.30 shows an overview of the proposed model.

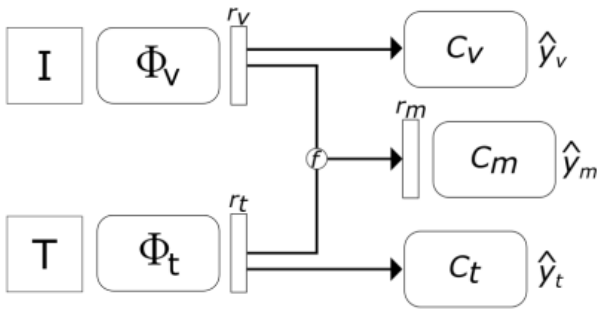


Figure 2.30: Overview of the proposed multimodal multi-task approach. (Source: image from [FCd19]).

The model is composed by:

- **Two features extractors:**
  - **Visual Network (Φ_v):** aims to extract a representation of the image;
  - **Textual Network (Φ_t):** aims to extract a representation of the text.
- **Three auxiliary classifiers:** each classifier specializes itself either in the classification of the image representation, the text representation, or the multimodal representation.
  - **Visual classifier (C_v);**
  - **Textual classifier (C_t);**
  - **Multimodal classifier (C_m).**

During training, if an instance is an image-text pair the whole model is updated. Otherwise, if an instance is an image or a text, only the corresponding features extractor and classifier are trained.
During test time, the prediction is done with the corresponding classifier.
During the training the model handle with three tasks, which are:

1. The prediction of the sentiment $\hat{y}_v$ from the visual information only;
2. The prediction of the sentiment $\hat{y}_t$ from the visual information only;
3. (main task) The prediction of the sentiment $\hat{y}_m$ from the fusion of the image and text representation.

For each of the three tasks j ∈ {v, t, m}, its auxiliary loss $L_j(\hat{y}_j, y)$ is defined by the cross-entropy. The whole model is trained to minimize:

$$L = \alpha_v L_v(\hat{y}_v, y) + \alpha_t L_t(\hat{y}_t, y) + \alpha_m L_m(\hat{y}_m, y) \quad (2.1)$$

When an image-text pair is available, the two features extractors Φ_v and Φ_t and the three classifiers C are trained according to the multi-task loss defined in the equation 2.1.
When the instance is an unpaired image, the loss function only includes the prediction of the visual classifier:

$$L = \alpha_v L_v(\hat{y}_v, y) \quad (2.2)$$

The same happens when the instance is an unpaired text, the loss function only includes the prediction of the text classifier:

$$L = \alpha_t L_t(\hat{y}_t, y) \quad (2.3)$$

The authors used the following datasets for the experiments:

- **Flickr emotion:** only examples where the majority of workers agreed for a particular label were used.

  - 20% of examples that received all the votes for the same emotion were randomly divided equally to form the validation and the test sets.

- **VSO:** the authors downloaded the images and used the Flickr API to collect the texts associated with the images.

  - Samples with less than 5 words and more than 150 words were removed - resulting in 301,042 pairs of images and texts;
  - 80% of the images composed the training set;
  - 20% of the images composed the validation and the test sets, 10% each.

The authors made two experiments, which were compared with six variants as baselines:

- **Image-based classifier (SI):** Visual network with visual classifier;
- **Text-based classifier (ST):** Textual network with textual classifier;
- **Single-task multimodal classifier (SM):** Visual network and Textual network with Multimodal classifier;
- **Late-fusion:** Trained Image-based classifier and text-based classifier are reused and their predictions are averaged;
- **Multimodal-text only (SM_T):** Single-task multimodal classifier is reused but the images are absent at test time;
- **Multimodal-image only (SM_I):** Single-task multimodal classifier is reused but the texts are absent.

The authors evaluated the advantages of their multi-task approach on the generalization of each three tasks: text, image, and multimodal classification. The authors concluded that their model is more robust to a missing modality. However, it is interesting to see the difference between the results with FlickrEmotion and VSO. With FlickrEmotion it is possible to notice the advantage of multi-task learning, whereas with VSO late-fusion and the single-task multimodal classifier obtained similar performance with their model, the authors speculated that since VSO is noisy, the upper-bound that any algorithm can approach on this dataset is relatively low. On the other hand, since the dataset is very large, it is possible that simple models are already able to obtain relatively good performances. Figure 2.31 shows the results of Experiment 1 presented by the authors.
The authors evaluated the possibility of leveraging monomodal examples to improve multimodal classification. This was particularly useful for FlickrEmotion dataset which is limited to 8,163 image-text pairs, but actually contains 13,912 image-only examples. Thus, they trained their model with this additional data. The results show a significant improvement (5%) in comparison with the first experiment. However, this can be intuitively associated with the fact that now the visual classifier was trained with more data, and thus the quality of the visual representation is improved.

Method	Classifier	Accuracy	F1-Macro
Baselines	SI	70.59	0.5808
	ST	83.70	0.7982
	L-Fus	89.09	0.8564
	SM	89.93	0.8659
	SM_I	47.92	0.4064
	SM_T	79.90	0.7486
Ours	C_v	70.34	0.6067
	C_t	83.99	0.8095
	C_m	**91.17**	**0.8803**

a)

Method	Classifier	Accuracy	F1-score
Baselines	SI	69.91	0.7767
	ST	84.15	0.8779
	L-fus	85.73	**0.8913**
	SM	85.79	0.8868
Ours	C_v	69.73	0.7648
	C_t	83.79	0.8775
	C_m	**86.35**	0.8894

b)

Figure 2.31: Results of Experiment 1 presented by the authors. Table a) shows the results obtained with FlickrEmotion dataset. Table b) shows the results obtained with VSO dataset. (Source: image from [FCd19]).

With VSO dataset, a given fraction of the training set was kept as multimodal data, while the remaining portion was divided in two: the texts are removed from the first half and the images from the second. The advantage of performing multimodal classification is only visible when the model can be trained with a large amount of image-text pairs. Otherwise, when their model can also be trained with monomodal data, the multimodal classifier always performs better than a text-based classifier even with very few training image-text pairs.

Figure 2.32 shows the results of Experiment 1 presented by the authors.

Training	Classifier	Accuracy	F1-Macro
Multitask + images	C_v	75.47	0.6507
	C_t	83.92	0.8063
	C_m	**91.59**	**0.8810**

a)

b)

Figure 2.32: Results of Experiment 2 presented by the authors. Table a) shows the results obtained with FlickrEmotion dataset. Figure b) shows the results obtained with VSO dataset. (Source: image from [FCd19]).

The idea of handling missing modalities is interesting and useful, because in real environments seldom we will have both information available. Thus, the experiments show that not only their approach offers a viable and simple solution to a missing modality, but also that multitask learning can improve generalization. Thus, the approach of multi-task is

in the sense that it considers a multimodal and two monomodal classification problems at the same time. Some advantages of the multi-task approach are presented:

- They concluded that multimodal classification can generalize better compared to when each classifier is trained individually;
- It becomes easy to train a multimodal model with additional monomodal data.

The authors complain that sentiment analysis and emotion recognition are harder tasks with images than with texts. However, they don't talk about analyzing the sub-images in the entire image, which can provide significant information and define the conveyed emotion/sentiment, as well as using a facial expression recognition technique in images with faces.
Also, if they concluded late-fusion and the single-task multimodal classifier obtained a similar accuracy with their model, they could investigate employing one of these methods (because they are more simple), and evaluate the computational cost. Unfortunately, no information about computational cost was presented in the paper.
It would be interesting to see a work handling with ISA using the K-score to evaluate the performance of the model, and this could be more interesting with a multi-task model. However, the use of F1 was right due to the existence of uneven class distribution.
The authors could test their model behavior with a balanced class distribution. To extract the representation of the input image, the authors used a DenseNet121 pretrained on ImageNet, they could have evaluated their model with other pretrained models and the respective execution time.
The work [OFB20] presents a review of the most relevant works in Visual Sentiment Analysis, which were published between 2010 and 2019 (27 works). Thus, this work aimed to be a guide for researches that are interested in Visual Sentiment Analysis. The authors presented two approaches to emotion modeling:

- **Dimensional approach:** emotions are represented as points in a 2 or 3-dimensional space, which have three basic underlying dimensions: valence, arousal, and control;
- **Category approach:** there is a number of basic emotions.

Of all the models that the author presented, the only one that is not included in the previous report is the Mikels' model, which defines that there are 8 basic emotions, which are: amusement, awe, disgust, contentment, anger, excitement, fear, and sad. Although the article [OFB20] is from 2020, the author does not mention Russel's Circumplex Model/1980 (which represents the 2D dimensional approach, considering only the arousal and valence axes), Parrots' Classification of Emotion Model/2001 (which classifies emotions in a tree-structured, so that there are six primary emotions, and secondary emotions are derivations and not junctions of primary emotions), and Hugo Lövheim Cube of Emotions/2011 (a dimensional approach, however, it does not consider arousal, valence, and control, but rather agrees with the activation of neurotransmitters).
The authors arguing that the most popular model is Plutchik's Wheel of Emotions, which is a well established psychological model of emotions.

The author describes that a factor that makes it difficult to compare techniques and approaches is the fact that the choice is arbitrary, i.e., it is difficult to compare a model that uses a pre-trained CNN and in the end, it will give us one of the 24 Plutchik's categories and a model that combines textual and visual information. However, it's possible to observe that until 2016 the models were mostly based on hand-crafted visual features, however in 2019 it started to use models that combine the image with the associated metadata. The author emphasizes that there is no correct strategy, but lately, CNN and multimodal approaches to learning representation have been shown to be promising, and capable of incorporating the contribution of multiple sources of information.
It would be interesting if the author could have mentioned pros and cons regarding the use of traditional and deep-learning algorithms, if the authors of the works presented the training and test times, one of the comparisons that could be made was based on temporal performance.
The authors talk briefly about some datasets, which can be divided into two types:

- **Psychological based:** this kind of dataset requires high efforts to be built and maintained over time. Thus, it is not possible to build large scale datasets;
- **Social media based:** datasets of images shared through social media, provide an easy way to retrieve images on a very large scale. However, due to the number of images, it's necessary to use automatic labeling, and as consequence may cause unreliable labels (metadata includes sarcasm, personal considerations).

One of the most difficult steps for the design of a Visual Sentiment Analysis model is the selection of the data features that better encode the information that the model is aimed to infer. The authors presented the three categories that the image features can be divided into:

- **Low-level features:** minor details of the image, like color histograms, texture, color, lines or dots, that can be picked up by, Scale-Invariant Feature Transform (SIFT), GIST, or HOG;
- **Mid-level features:** these features bring more semantic. They are learned using class-level information are potentially more distinctive than the traditional low-level local features, and are more interpretable, and have stronger associations with emotions;
- **High-level features:** are built on top of low-level features to detect objects and larger shapes in the image, i.e these features describe the semantic concepts shown in the images.

As shown in Figure 2.33, the most used approach involves raw images (15/27), with or without the combination of the raw image with the associated metadata. In the last, the features taken from different modalities are combined to create a common vector space. Using the raw image, the system automatically learns how to extract the needed information from the input data (e.g, training a CNN). However, these methods require huge amounts of labeled training data and an intensive learning process.

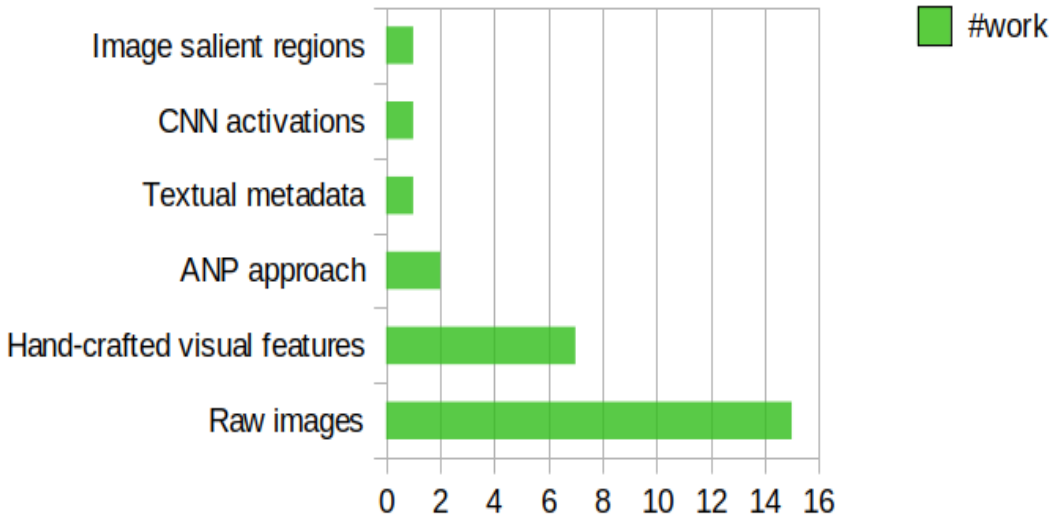


Figure 2.33: Distribution of approaches used by the studied works in [OFB20].

Another approach often used (7/27) is the combination of a huge number of hand-crafted visual features. However, there is no agreement about which of them gives a major contribution to the aimed task.
There is no agreement about which strategy to select. However, according to the authors, recent results suggest that it is worth investigating the use of representation learning approaches (e.g CNN) and multimodal embedding that can embody the contribution of multiples sources of information [ZWS+20].
The authors mentioned some additional challenges and techniques that can be investigated, the most relevant to our research are:

- **Relative attributes:** Given a set of images that have been assigned to the same emotional category, the authors suggest that would be interesting to determine their raking concerning the specific attribute. It's possible to see an approach like this in work [GA19a], which has a probability distribution table to the item (in an image) owing to a certain category;

- **Common sense:** The reduction of the affective and cognitive gap between images and sentiment conveyed by them. For example, a Halloween picture can be classified as a negative image, however, the knowledge of the context should affect the semantic concepts conveyed by the picture. The authors suggest the exploitation of the Attention mechanism, which makes the Artificial Neural Network work better by letting the network know where to look as it is performing its task;

- **Emoticon/Emoji:** possibility to exploit text ideograms, since this was introduced to allow the writer to express feelings and emotions concerning a textual message.

It would be interesting if the authors presented the most used metrics to evaluate the models, and also going a little deeper about the use of facial expressions recognition to help the

sentiment prediction (images with faces), and problems like the bright colors and bright areas in negative images, that influence a wrong classification.

The work in [STD14] aims to present the results of recognition of seven emotional states (neutral, joy, sadness, surprise, anger, fear, and disgust) based on facial expressions.
The research was made based on the fact that light conditions and changes of head positions are the main factors that affect the quality of emotion recognition systems using cameras.
For their study, the authors used Microsoft Kinect for 3D face modeling, which has an infrared emitter and two cameras. One of the cameras record visible light, while the other operates in infrared and is used for measuring the depth. The model is based on 121 specific points of the face, which are arranged on characteristics positions on the face. Then, the spatial coordinates of the points are stored in a form of a matrix. The authors also take advantage of the use of Kinect since the device provides six AUs derived from the Facial Action Coding System (FACS) system (changes in facial expressions resulting from the activity of specific muscles, which were organized in the form of special coefficients).
For the material, the authors took six men aged 26-50 to conduct the experiment. A participant task was to play mimic effects according to instructions on a computer screen. The instructions presented the name of the emotional state and a picture (from KDEF database [CL08]) of an actor performing the corresponding mimic effect, in order to make it easier for the participant to reproduce the emotion. In the end, the authors created an entire database, which contains 252 facial expressions.
For the classification process, six AUs were used as features, which are:

- **AU0:** upper lip raising;
- **AU1:** jaw lowering;
- **AU2:** lip stretching;
- **AU3:** lowering eyebrows;
- **AU4:** lip corner depressing;
- **AU5:** outer brow raising.

It is possible to organize these AUs in a distribution table for each emotion. Figure 2.34 shows the table presented in the work.
The authors conducted their experiments using kNN and Multilayer Perceptron (MLP) classifiers, in order to test the possibility of automatic recognition of emotions using AUs. The authors used a 3-Nearest Neighbors (NN) classifier and two-layer neural network classifier (MLP) with 7 neurons in the hidden layer. Figure 2.35 shows the respective structure of the neural network.
The authors tested two ways to recognize emotions: a) subject-dependent; b) subject-independent, for both, the data were randomly divided on teaching (70%) and testing

ES	neutral	joy	surprise	anger	sadness	fear	disgust
							
AU0	0.21	0.77	-0.10	0.30	0.17	-0.11	0.91
AU1	-0.06	0.09	0.60	-0.07	-0.04	0.20	0.13
AU2	-0.25	1.00	-0.49	0.06	-0.37	-0.60	0.88
AU3	-0.21	0.00	-0.13	0.04	-0.09	-0.17	0.00
AU4	-0.04	-0.47	0.58	-0.19	-0.02	0.28	-0.32
AU5	-0.23	-0.30	0.10	-0.34	-0.27	-0.02	-0.39

Figure 2.34: Table presented in [STD14], which presents the facial expression and the corresponding action units distribution.

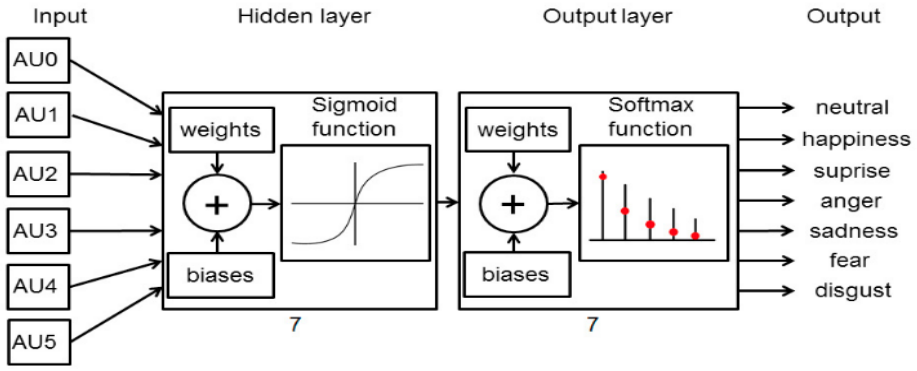


Figure 2.35: The neural network architecture for the proposed method. (Source: image from [STD14]).

(30%) for 3-NN classifier, and for MLP into three groups: teaching (70%), validation (15%) and testing (15%).
The authors could conclude that for recognition of emotions based on facial expressions for all users is much more useful and versatile than for an individual user. In subject-independent approach, the classifier accuracies (CA) for 3-NN and MLP algorithms were respectively 95.5% and 75.9%. However, for subject-dependent the results obtained were 90% for MLP classifier, and 96% for 3-NN. The authors generated the confusion matrices in order to identify which emotions are the easiest and which the most difficult to distinguish. The confusion matrix for 3-NN classifier presented low values for samples that were wrongly classified. However, the confusion matrix for MLP classifier presented higher values for samples wrongly classified, for example, for sadness, the model resulted 505 neutral as false positive.
The authors also tested the different division of the data (learning and testing). For subject-dependent classification, the authors divided the data into 6 subsets with all 7 facial expressions - five subsets were used for teaching and one for testing. The results presented 73% accuracy for MLP classifier, and 70% for 3-NN.
For subject-independent classification, the authors divided the data into 12 subsets - eleven subsets were used for teaching and one for testing. The results presented 73% accuracy for MLP classifier, and 63% for 3-NN.
The authors concluded that neural networks have a good ability to generalize. Also, through the confusion matrices, it was possible to identify the model behaviour. Therefore, they concluded the most difficult to recognize were: sadness and fear, which were often confused respectively with neutral and surprise emotions. This is probably caused by using only six AUs.

The work in [LGAC21] proposed a novel multi-modal model, which will use both textual data and images from social media to perform the classification: positive, neutral, and negative. The respective model consists of initial classification of the textual and image components, and then fuse both classifications into a final one using an Automated Machine Learning (AutoML) approach, which will perform a random search to determine the best model to perform the final classification.
The proposed method was evaluated using a dataset containing over 470,000 tweets, where each tweet is composed of both textual and image content.
The proposed method is composed of three stages, which are:

- **Pre-processing:** the first container represents the pre-processing component, that receives an image and associated text, and pre-processes it to remove noise and non-important data;
- **Individual classifications:** the second container shows both classification components, where the image and the text are classified individually using CNNs;
- **Fusion stage:** the third container receives the concatenation of the individual classifications, and performs a final classification using the optimal model searched -

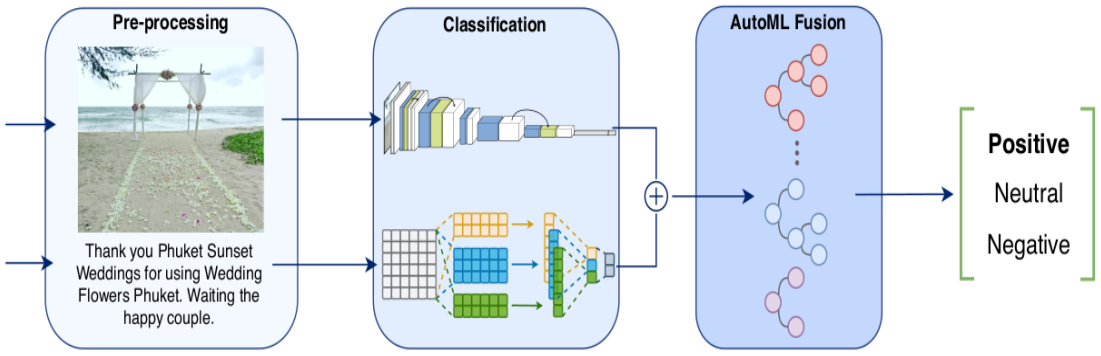


Figure 2.36: Proposed multi-modal architecture. It's possible to identify 3 main components on the architecture. (Source: image from [LGAC21]).

represented by a Gradient Boosting Machine (GBM) in the image.

For the image sentiment analysis, the authors explored the use of state-of-the-art CNNs that perform feature extraction and classification, they focused on two architectures: ResNet and DenseNet. The images were resized (224 x 224), in order to uniform the dataset, then a normalization was applied on each image, which was obtained by subtracting the mean and dividing by the standard deviation in each channel of each image.

For ResNet, the authors implemented ResNet18, ResNet34, ResNet50, ResNet101 and ResNet152, using the parameters defined in the original paper [HZRS16]. For DenseNet the authors implemented DenseNet161 using the parameters detailed in the original paper [HLVDMW17].

For the text sentiment analysis, a pre-processing step was applied to the input text in order to clean the textual data. Thus, the steps were: i) transform HyperText Markup Language (HTML) codes into words and symbols; ii) remove stop words using Natural Language Toolkit (NLTK) functionalities; iii) transform every word to lower case; iv) remove occurrences of more than three equal sequential characters into a maximum of two; v) remove links and specific social media user-mentions (both the mention and the "RT" word from Twitter); vi) punctuation was removed.

For the fusion method, the authors focused on using the individual classifications of the text and image components to perform a final classification. The method was based on AutoML, in order to create an optimal model to classify a given dataset, without requiring extensive human modelling. To search for the optimal machine learning model, the authors based their solution by performing an automatic random search [BB12] over a set of several machine learning algorithms and their inner parameters. The search was performed over a space that includes 2 models: i) a random forest; ii) a random grid of XGBoost. After searching, 2 stacked ensembles were created, the first one comprised of all models evaluated, and the other one, containing the best model of each type. In the end, the model with the best performance on the validation set is the one selected to be in the architecture of the proposed method.

The authors used the B-T4SA dataset. Furthermore, the authors incorporated two more datasets: Stanford Sentiment Treebank (SST)-5, for the text classification, and Flickr and

Model	Approach
TextBlob	Values under –0.1 are classified as having a negative meaning, over 0.1 are classified as positive, and the remainder is classified as neutral.
FastText	An Embedding layer followed by two Linear layers, the first having input size equal to the embedding size and output equal to 256, whilst the second one has input size of 256 and outputs the classification vector.
LSTM	an Embedding layer, followed by an LSTM layer with an input size equal to the dimension of the Embedding and with the number of features in the hidden state equal to 256. This is then followed by a Linear layer with input size equal to the number of features in the LSTM (256) and output size equal to the number of classes.
LSTM-Attn	An Embedding layer, followed by an LSTM layer with input size equal to the embedding dimension and 256 as the number of hidden features. Following this, comes the Attention layer that receives the output from the LSTM and the last hidden LSTM state, and outputs a new hidden state with the same size as the output of the LSTM. This then serves as input to the Linear layer, which outputs the classification vector.
Bi-LSTM	An Embedding layer, followed by a bidirectional LSTM layer with input size equal to the embedding dimension and 256 as hidden features. Then, the output from the Bi-LSTM layer performs both an average pool and a max pool, which are concatenated together and fed into a Linear layer of input size 256 x 4 and output of 64. Then, a ReLU operation is performed, followed by a dropout, with p = 0.1. The result of this goes to a Linear layer that outputs the classification vector.
RNN	An Embedding Layer followed by a multi-layer Elman RNN with 2 layers, input size equal to the dimension of the embedding and with a hidden size of 256. The output of this layer is then inserted into a Linear layer that outputs the classification vector.
R-CNN	An Embedding Layer, a bi-directional LSTM Layer with input size equal to the dimension of the embedding, hidden size of 256 and a dropout of 0.8. The final embedding vector is the concatenation of its embedding and left and right contextual embeddings, which in this case is the hidden vector of the LSTM. This concatenated vector is then passed to a Linear Layer which maps the input vector back to a vector with a size equal to the hidden size of the LSTM, 256. This is passed through a 1D Max Pooling Layer, and finally, the output from this layer is sent to a Linear Layer that maps the input to a classification vector.
TextCNN	a second CNN-based network was defined to perform text classification. The architecture has 3 with kernel sizes of 1, 3 and 5 respectively.
Very Deep Convolutional Networks (VDCNN)	It was implemented 4 VDCNN architectures with different depths: 9, 17, 29 and 49. Every architecture starts with an Embedding layer, followed by a 1D Conv layer with input size equal to embedding size and output size of 64. Then, they have a set of Convolution Blocks. Then comes a K-Max Pooling layer, a Linear layer with input of 512 x k, where k is the number selected for the pooling layer, and output of 2048. Following it, a Linear Layer with 2048 as input and output is inserted and finally, a Linear Layer with an input size of 2048, outputs the classification vector.

Table 2.11: Models evaluated by the authors and the respective application given to each one.

Instagram Dataset, for the image classification component to conduct experiments using transfer-learning [PY10].

To select the best text sentiment analysis model to use in the multimodal architecture, the authors conducted a set of experiments with all the models implemented, where each model was evaluated three times in the task of classifying sentiments in the B-T4SA dataset, using the Adam optimizer [KB14], and the Cross-Entropy loss.

From the results, the authors selected R-CNN as the text classifier to be used in the proposed method, since it presented the best results in the validation set (94.61%).

To select the best image sentiment analysis model, the authors conducted a few experi-

ments.
From the obtained results, it was possible to see that all models performed similarly, but ResNet34 was the best one, achieving 49.8% accuracy using RGB images, with or without pre-training. Even though ResNet18 had almost the same performance using pre-trained settings, the authors selected ResNet34 for the proposed method, as it consistently outperformed ResNet18.
After selecting the models, the authors evaluated the performance of the proposed method as a whole. They conducted the experiments using a baseline using SVM, and the AutoML-based Fusion. The difference in the results was small (SVM - 95.16%, AutoML - 95.19% - 0.03%), but the authors selected the AutoML due to several advantages, one of them it's the time required to train, while AutoML method requires two hours, SVM required several hours to train.
Finally, the authors validated their model comparing the results with the state-of-the-art methods. From the results, the authors could conclude that their method was capable of finding an optimal model that outperformed the state-of-the-art in the B-T4SA dataset, with 95.19% (using AutoML), which, due to its natural content, is very challenging and contains intra and inter-class subjectivity.

To summarize all the information obtained during the survey of the state of the art, Table 2.12 presents some of the recent works, relevant for this project, that can be found in the literature, which aim to propose models to handle the image sentiment analysis.

We notice that, recently, several works employed models with image and text classification. However, none of them employed an approach that could handle images, salient regions, textual data, and facial expressions. Regarding the final output, over 85.71% employed polarity as the final classification, using either 2 classes (positive and negative), or 3 classes (positive, neutral, and negative). Only one work employed an approach using an emotional model.

2.8 Conclusion

This chapter presented an overview of the image sentiment analysis area. We presented the most common datasets, the sentiment models, a brief overview of traditional machine learning approaches and also deep learning approaches to image sentiment analysis. We also focused facial emotion recognition and discussed object detection. It was possible to identify the available tools and technologies, which can be used in our own work, and using the knowledge acquired during the study of the area, it will be possible to propose a method that can innovate and improve the state-of-the-art in image sentiment analysis.

Work	Year	Proposed Model	Global Image	Salient Regions	Face	Text	Polarity	Emotion Model	Dataset
[YLJY15b]	2015	PCNN	X	-	-	-	2	-	Flickr,and Twitter-I
[CSGiNJ15]	2015	CNN	X	-	-	-	2	-	Twitter (created by You)
[SYWS16]	2016	CNN	X	X	-	-	2	-	Flickr, and Twitter-I
[YSS17]	2017	Augmented Conditional Probability Neural Network (ACPNN)	X	-	-	-	-	Mikels'	Emotion6, Twitter-I, and Abstract (from AIC)
[VCC+17b]	2017	Multimodal model	X	-	-	X	3	-	T4SA/B-T4SA, and one created by the authors
[SYLM18]	2018	CNN w/ visual attention	X	-	-	-	2	Mikels'	Twitter-I, and ART-photo
[OFTB18]	2018	Multimodal model	X	-	-	X	2	-	Flickr
[WQJZ20b]	2019	Multimodal model	X	X	-	-	2	-	Flickr, FI, Twitter-I, Twitter-II, and EmotionROI
[FMP+19]	2019	Deep Convolutional Neural Network (DCNN)	X	-	-	X	3	-	SocIal MediaPictureS News-related (SIMPSoN)[sim19]
[HZZ+19]	2019	Deep Multimodal Attentive Fusion (DMAF)	X	-	-	X	2	-	Getty, Twitter-I, and Flickr
[GA19b]	2019	Multimodal model	X	-	-	X	3	-	B-T4SA
[GA19a]	2019	Evaluation of [GA19b]	X	-	-	X	3	-	B-T4SA
[ZWS+20]	2020	Cross-modal sentimental semantics	X	-	-	-	2	-	Twitter-I and FI
[FCd19]	2020	Multimodal model	X	-	-	X	-	-	Flickr, and VSO

Table 2.12: The works that can be found in the ISA's literature and the respective approaches.

Chapter 3

Proposed Method and Implementation

3.1 Introduction

This chapter presents the proposed method and its components. This chapter is split as follows: the section 3.2 presents the proposed method and a brief discussion about its components and the tasks to be accomplished for each model. The following section 3.3 presents the implementation of the facial expression recognition model, with a brief discussion. Section 3.4 presents the implementation of the image classification model, with a brief discussion. Section 3.5 presents the implementation of the salient area classification model. Section 3.6 presents the implementation of the text classification model. The final section 3.7 contains the discussion about the junction of all models to obtain the final model and presents how it was done. Finally, the section 3.8 contains the main conclusions made while formulating this chapter.

3.2 Proposed Method Overview

According to the study made in chapter 2, it was possible to idealize a new approach to be developed. Figure 3.1 shows an overview of the proposed model that was developed during this project.

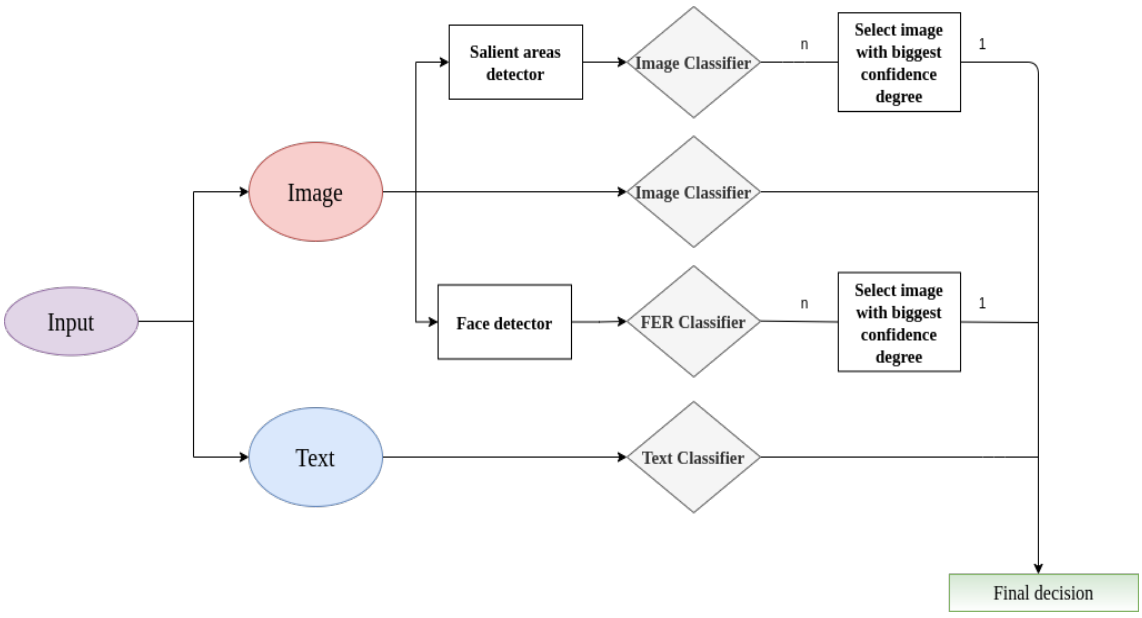


Figure 3.1: Overview of the proposed method's architecture and its components.

The proposed method fuses the outputs of all models. The model will support receiving an image and text (not mandatory), or a Comma-Separated Values (CSV) file containing

the paths for each image and the respective sentence to be analyzed.

The proposed method will be composed by 4 models, which are:

- **Image classifier:** model responsible for analysing the sentiment of the original image. This is a mandatory model, that is, the information returned by this model will be always considered. It will be used to process the full image and also its salient regions;
- **Facial Expression Recognition Classifier:** model responsible for analysing the sentiment of the identified faces on the original image. The global features can give hints regarding the feeling of the image, but several works faced difficulties with the model getting an erroneous classification due to global features. Therefore, the objective of using a model that performs the classification of facial expressions would be to address these problems that may be faced by the proposed method during classification. From the results returned by this model, the information to be considered will be the one that has been obtained with the highest confidence degree;
- **Salient areas detector:** component responsible for detecting the salient areas in the original image. The objective of using a model that performs the detection of salient areas would be to address the global features problems that may be faced by the model during classification, and it will allow to get a sense of which objects are contained in the images. Likewise the FER model, from the results returned by this model, the information to be considered will be the one that has been obtained with the highest confidence degree;
- **Text Classification:** even though this project does not include the development of a text classifier, in order to evaluate the proposed method's behaviour when exposed to the textual information, it is necessary to use one. To this end, several models were evaluated.

The following sections are responsible for presenting in detail each component, their roles, how they work, and a discussion regarding the chosen approaches.

3.3 Facial Expression Recognition

3.3.1 Proposed Facial Expression Recognition Model

The Facial Expression Recognition model is responsible for two tasks: i) detecting faces in the images; ii) classifying the detected faces' expressions.
To make the detection in the images, we use the MTCNN module from the facenet_pytorch library. This model has three convolutional networks (P-Net, R-Net, and O-Net).
When receiving an image, the model will create an image pyramid, in order to detect faces of different sizes. Then, it is possible to split the MTCNN operation into three stages [ZZLQ16]:

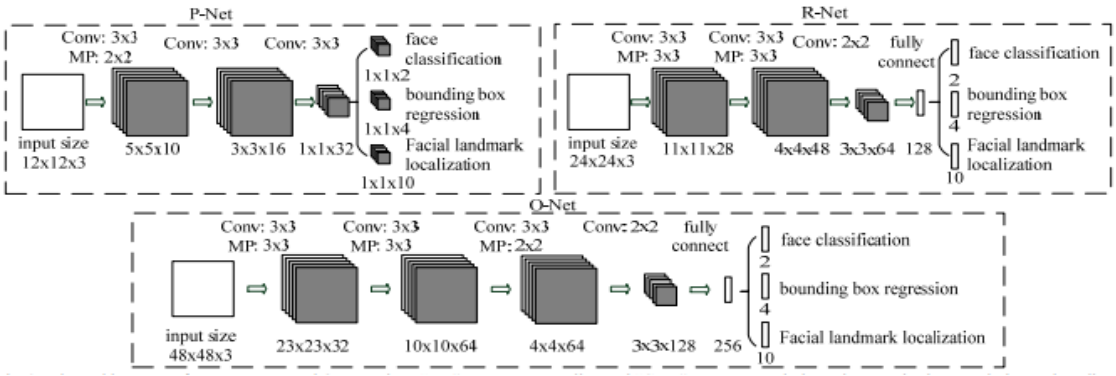


Figure 3.2: Overview of MTCNN structure. The architectures of P-Net, R-Net, and O-Net. The step size in convolution (Conv) and pooling (MP) is 1 and 2, respectively. (Source: image from [ZZLQ16]).

- **Stage 1:** : A fully convolutional network is exploited (P-Net), in order to obtain the candidate facial windows and their bounding box regression vectors. Then, candidates are calibrated based on the estimated bounding box regression vectors. Then, Non-Maximum Suppression (NMS) is employed to merge highly overlapped candidates;

- **Stage 2:** All candidates are fed to another CNN (R-Net), which further rejects a large number of false candidates, performs calibration with bounding box regression, and conducts NMS;

- **Stage 3:** This stage is similar to the second stage, but this stage is aimed to identify face regions with more supervision. In particular, the network will output five facial landmarks' positions, O-Net.

Additionally, its high accuracy is obtained by using deep neural networks. Having three networks allows producing a higher precision since each network can fine-tune the results of the previous one. The model also employs an image pyramid to find faces both large and small. Even though this may provide an overwhelming amount of data, NMS, as well as R-Net and O-Net, all help discard a large number of false bounding boxes.
Since it is able to deliver high accuracy with less run-time, these qualities make MTCNN one of the most popular and most accurate face detection tools.

Therefore, the FER model function will receive an image, which will be converted to an array and then, using the MTCNN, the bounding boxes will be obtained, and also the confidence degrees of each detected face in the image.

Then it will be checked if any face was detected in the image since there is a possibility that there are no faces in the image, that is, the list of bounding boxes has returned empty.
Otherwise, for each identified bounding box, the values X, Y, w (width), and h (height) will be obtained, in order to perform the clipping of the respective face in the image. Before saving the crop, the resolution of the image will be checked, in order to maintain a certain level of quality of the images obtained and prevent images of very low quality (and which

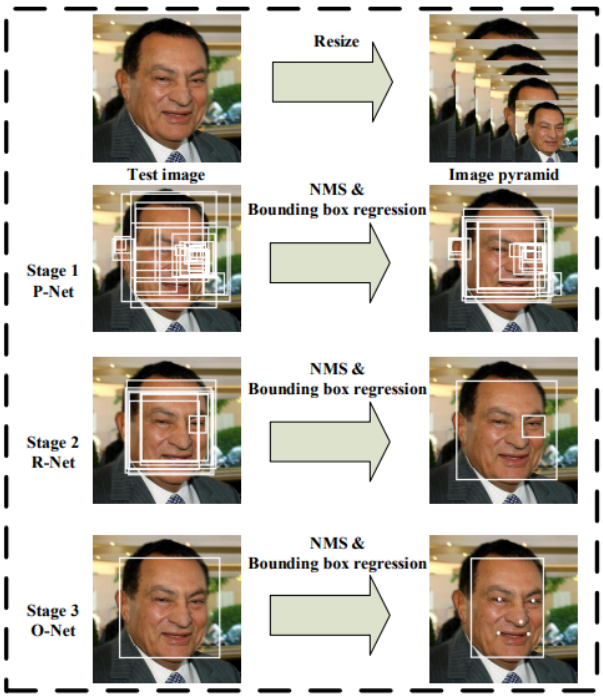


Figure 3.3: Pipeline of the cascaded framework that includes three-stage multi-task deep convolutional networks. Firstly, candidate windows are produced through a fast P-Net. After that, these candidates are refined in the next stage through an R-Net. In the third stage, the O-Net produces the final bounding box and facial landmarks position. (Source: image from [ZZLQ16]).

would not add any utility to the model) from being kept.

The rule of only saving images with a resolution greater than or equal to 16x16 pixels was added. However, bad quality images were obtained. Figure 3.4 presents one of the images obtained in one of the execution tests.

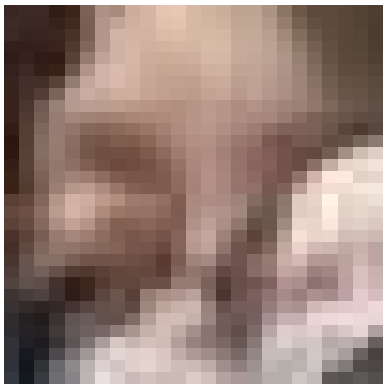


Figure 3.4: Example of image that were obtained with the rule setup to save images with resolution greater than or equal to 16x16.

Therefore, the rule was changed to only save images with resolution greater or equal to 30 x 30. Figure 3.5 presents an image demonstrating the results obtained from the dataset built with images from Twitter, and with the new rule. Then, these images were saved and their paths were added to a list, which would be processed to obtain the emotion classification.

After the face detector, the emotion recognition is performed. For this task, a CNN was

Figure 3.5: Results obtained with the new rule.

created with ResNet9, which increases (gradually) the number of channels of facial data and decreases the dimension, followed by a fully connected layer responsible for returning an array with 7 values between -1 and 1, describing the probability of the class belonging. The learning rate scheduler, 1Cycle, was implemented so that the learning rate was not manually implemented. It starts with a very low learning rate, increases, and again decreases. Figure 3.6 shows the model's architecture.

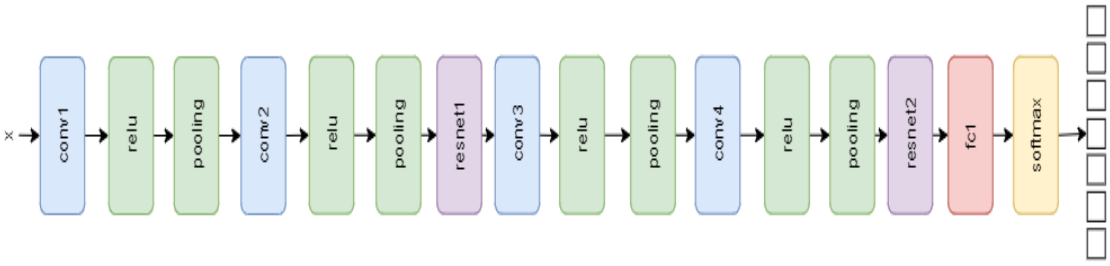


Figure 3.6: Architecture of the FER model.

During the tests, it was possible to observe the advantage of having increased the dataset size, taking into account the value initially obtained in test_0, and even in the values obtained between tests B and D. In this way, the model was saved in order to be used later in the proposed method with all the models.
After the training and configuration phase of the FER model, it underwent some changes to be adapted for the pipeline junction:

- The model received the list of faces (obtained by the face detector), to would be classified;
- A Softmax layer was added in order to normalize the values of degree of confidence in the interval [0,1];

- The 7 classes were converted to 3, in order to maintain the consistency of the classes.

3.3.2 Datasets for Facial Expression Recognition

For training, the FER2013 dataset was used. This dataset consists of 48x48 pixel grayscale images of faces. The faces have been automatically registered so that the face is more or less centred and occupies about the same amount of space in each image. The labels are divided into 7 types: 0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral. The training set consists of 28,709 examples and the validation and test sets consists of 3,589 examples each. The model achieved approximately 68.82% accuracy in the test set with data from FER2013.
However, a set of data was prepared with social media images, namely Twitter, in order to assess the model's behavior when exposed to social media images, which may or may not have large resolutions. The accuracy obtained with the Twitter image test data set was approximately 18.22%. Figure 3.7 shows the confusion matrix obtained.

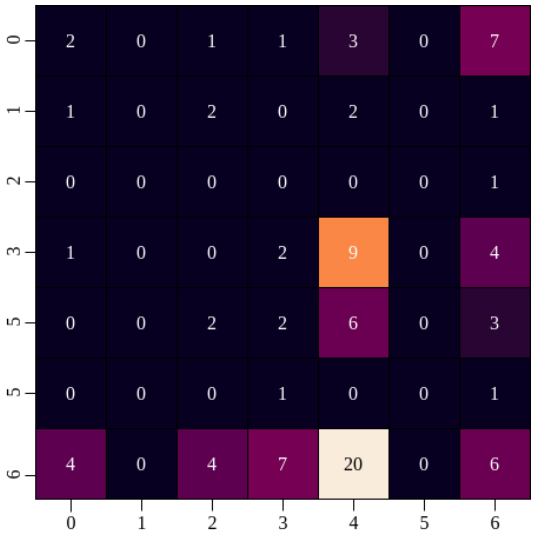


Figure 3.7: Confusion matrix obtained from the dataset with images from Twitter, used in order to identify the behavior of the model with images from social networks.

Observing the low accuracy obtained with the set of tests with Twitter images, it was planned to increase the FER2013 dataset so that the model could have contact with data that were closer to the test images (images from social networks).
A dataset with a wide range of images was found so that it was possible to recognize even images from the dataset JAFFE (images made in laboratory environments). So we will call it Mixed dataset. This dataset contains a total of 13,691 images, which are labeled. Figure 3.8 presents one of the images that make up the Mixed dataset. It is possible to see that it contains images that were not obtained in laboratories.
However, it was possible to verify the need to use images directly taken from Twitter, since the posted images are often of low quality.

In this way, the training images from the dataset previously created (to train the object detector) were used. However, the images were not organized for the FER model, that is,

Figure 3.8: Example of an image that is part of the Mixed dataset, so that it is possible to notice that it was not obtained in a laboratory.

the images had much more information than just the faces. Therefore, the face detection function was used in order to only capture the necessary information from the images, that is, the faces.

The second step was to tag the images. It was based on the classes used in the FER2013 dataset, that is, Anger, Disgust, Happy, Sad, Surprise, Fear, and Neutral. The labeling phase is a little challenging, mainly because we have no specialization in the analysis of facial expressions, and some expressions can lead to confusion, such as fear and surprise.

Therefore, for the first two tests, only the data from FER2013 and TwitterFER were used, containing: Training - 29,676 images, Validation - 3,709 images, and Test - 3,710 images.

Figure 3.9 presents one of the images that compose the dataset used.

Figure 3.9: Sample images of the dataset (FER2013 and TwitterFER) used. Sentiment from left to right: surprise (5), sad (4), happy (3), angry (0), and neutral (6).

Finally, for the last two tests, all the datasets were concatenated, resulting in a final dataset with 50,783 images, which were divided into: 40,627 samples for training, 5,078 samples for testing, and 5,080 samples for validation.

3.3.3 Model Evaluation

3.3.3.1 Test A - FER2013_TwitterFER

For the first test, the number of epochs was set to 150 in order to assess the behavior of accuracy and loss.
The training had a total duration of 20 minutes and obtained an accuracy of 69.28% for validation. Figure 3.10 presents the accuracy values as a function of the number of epochs.

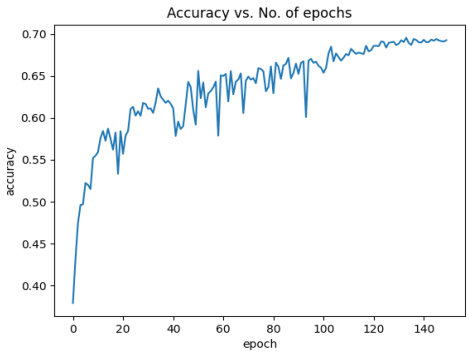


Figure 3.10: The graph obtained from the execution of the training, which shows the values of accuracy (validation set) as a function of the number of epochs.

Figure 3.11 presents the graph, which contains information on the value of the learning rates as a function of the number of batches.

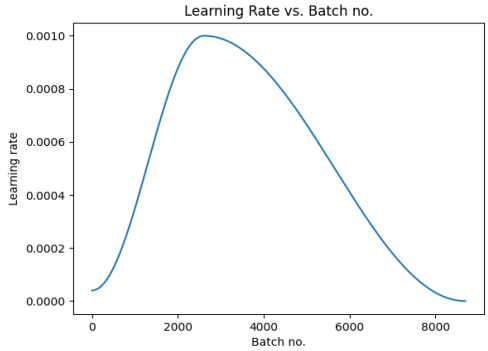


Figure 3.11: The graph obtained from the execution of the training in 150 epochs, which shows the values of the learning rate as a function of the number of batches.

As mentioned, the model makes use of the OneCycleLR scheduler, which defines the learning rate for each group of parameters according to the 1 cycle learning rate policy. The 1 cycle policy changes the learning rate from an initial learning rate to some maximum learning rate and then from that maximum learning rate to some minimum learning rate that is much lower than the initial learning rate.

Figure 3.12 presents the information of the loss value as a function of the number of epochs.
It is possible to have a notion that from the time of number 45 (approximately), the graph shows an overfitting behavior.

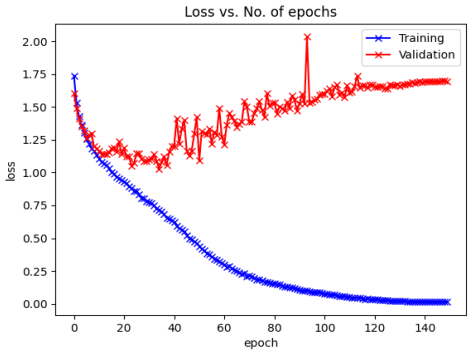


Figure 3.12: The graph obtained from the execution of the training in 150 epochs, which shows the Loss values as a function of the number of epochs.

Finally, the test presented an accuracy of 65.16%, whose value, taking into account the test values, with images from Twitter, presented initially in this subsection (approximately 18.22%), shows a significant rate of increase.

3.3.3.2 Test B - FER2013_TwitterFER

For this test, the number of epochs was configured taking into account the graph presented by Figure 3.12, in which it is possible to notice an overfitting behavior of the model from epoch 45 (approximately).

The training lasted 7 minutes and obtained an accuracy of 67.67% for validation. Figure 3.13 presents the accuracy values as a function of the number of epochs.

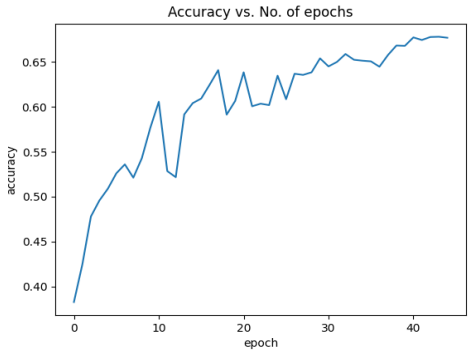


Figure 3.13: The graph obtained from the execution of the training, which shows the values of accuracy (validation) as a function of the number of epochs.

Figure 3.14 presents the loss value as a function of the number of epochs. Finally, the test had an accuracy of 64.72%.

3.3.3.3 Test C - FER2013_TwitterFER_MixedFER

For the first test with the final dataset, it was done similar to Test A, presented in subsection 3.3.3.1. The number of epochs was set to 150 in order to evaluate the behavior of

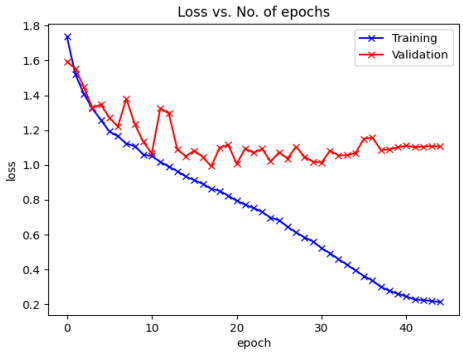


Figure 3.14: The graph obtained from the execution of the training in 45 epochs, which shows the values of loss as a function of the number of epochs.

accuracy and loss during the epochs since the number of images has been increased.

The training lasted 27 minutes and obtained an accuracy of 74.23% for validation. Figure 3.15 presents the accuracy values as a function of the number of epochs.

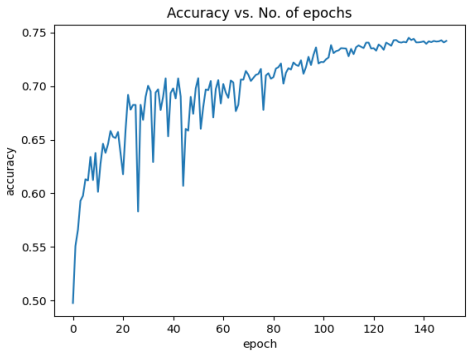


Figure 3.15: The graph obtained from the execution of the training, which shows the values of accuracy (validation) as a function of the number of epochs.

Figure 3.16 presents the loss value as a function of the number of epochs.

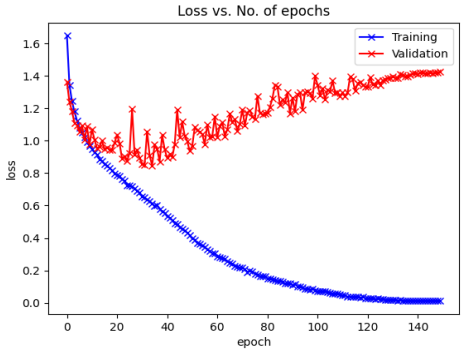


Figure 3.16: The graph obtained from the execution of the training in 150 epochs, which shows the loss values as a function of the number of epochs.

It is possible to observe that from the epoch 65 (approximately), the graph shows an over-fitting behavior. The highest loss value obtained by the validation was 1.4 and the lowest

was approximately 0.83.

Finally, the test showed an accuracy of 73.01% which, taking into account the test values presented in the subsection 3.3.3.1, shows a very pleasant increase rate, with an increase of approximately 12.05%.

3.3.3.4 Test D - FER2013_TwitterFER_MixedFER

Finally, the number of epochs was configured taking into account the graph in Figure 3.16, in which it is possible to notice an overfitting behavior of the model from epoch 55 (approximately).

The training lasted approximately 13 minutes and obtained an accuracy of 74.10% for validation. Figure 3.17 presents the accuracy values as a function of the number of epochs.

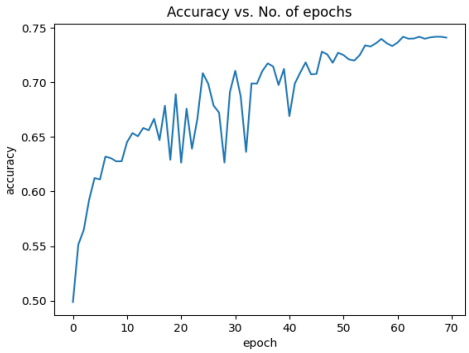


Figure 3.17: The graph obtained from the execution of the training, which shows the values of accuracy (validation) as a function of the number of epochs.

Figure 3.18 presents the loss value as a function of the number of epochs.

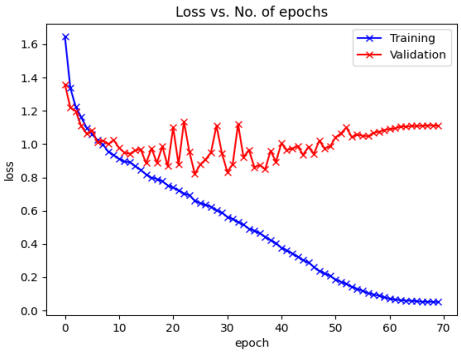


Figure 3.18: The graph obtained from the execution of the training in 55 epochs, which shows the values of loss as a function of the number of epochs.

Finally, the test showed an accuracy of 72.75%, an increase of approximately 12.41% over the value presented in the results obtained in the subsection 3.3.3.2. Figure 3.19 contains the respective confusion matrix.
Table 3.1 shows an overview of the obtained results.

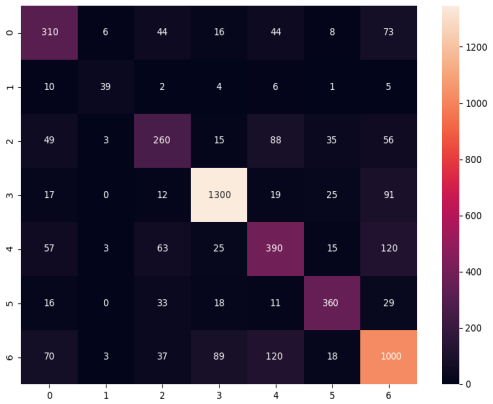


Figure 3.19: Confusion matrix obtained from the Test D results.

	Datasets				Validation		Test
	FER2013	**Twitter**	**Mixed**	**Epochs**	**Time(min)**	**Accuracy**	**Accuracy**
Test_0	X	-	-	-	-	-	68.82%
Test_0	-	X	-	-	-	-	18.22%
TestA	X	X	-	150	20	69.28%	65.16%
TestB	X	X	-	45	7	67.67%	64.72%
TestC	X	X	X	150	27	74.23%	73.01%
TestD	X	X	X	55	13	74.10%	**72.75%**

Table 3.1: Comparison between the results obtained from the tests.

The confidence produced by the model are normalized using a Softmax layer. The lists of the degree of confidence and predicted classes will be passed to an auxiliary function, responsible for converting each classification obtained by the model into one of the 3 classes (negative, neutral, and positive) and return the class that was predicted with the highest degree of confidence.

The classes are converted, as follows:

- Angry, Disgust, Fear,and Sad -> Negative;
- Surprise, and Neutral -> Neutral;
- Happy -> Positive.

Some research, before the FER model implementation, was made in order to find a good model that could be used for this project. Table 3.2 presents the models found that would be evaluated in order to find the best option, if any, to be used in this project.
However, taking into account that the objective is to implement as a model, it was not advisable to implement a complex model, since it would still be necessary to carry out training and tests. Therefore, it was decided to develop a model from scratch using a built dataset, which was presented in this section. It was possible to obtain good accuracy, even higher than the models presented in Table 3.2, however, with low complexity.

Method	Rank	# Classes	Accuracy
Pyramid With Super Resolution for In-the-Wild Facial Expression Recognition [VLYK20]	1	8	60.68%
Region Attention Networks for Pose and Occlusion Robust Facial Expression Recognition [WPY+19]	2	8	59.5%
Efficient Facial Feature Learning with Wide Ensemble-based Convolutional Neural Networks [SMW20]	3	8	59.3%
Compacting, Picking and Growing for Unforgetting Continual Learning [HTW+19]	7	6	63.57%
Facial Motion Prior Networks for Facial Expression Recognition [CWC+19]	9	6	61.52%

Table 3.2: Facial Expression Recognition models found that would be evaluated in order to find the best option, if any, to be used in this project.

3.4 Image Classification

For the image classifier, the architecture proposed in [GA19b] was used, since the proposed configurations obtained greater results than [VCC+17b].
A pre-trained ResNet152 network was used, with the last layer being fully-connected, accompanied by a Softmax layer, which will have 3 outputs, which will represent the probability of each class (negative, neutral, and positive), in the range [0,1], where 1 represents that the image belongs to that respective class and 0 that it does not. Thus, it was used a saved model, which was trained with the B-T4SA dataset.

In this way, the model will receive an image, which will be passed by the classifier, so that it will result in the respective predicted class and its respective probability, which can be seen as the degree of certainty with which that class was predicted.

3.5 Salient Areas Recognition

For the classification of salient areas, the image model will be used. However, it will not be sent from the whole image, but areas detected by the object detector, which will be cut and saved.
The detector chosen was YOLOv5, briefly described in the subsection 2.4.0.8. It was chosen due to being a recent launch and having demonstrated better values in comparison to other object detectors.
YOLO5 has been developed with PyTorch, and provides four models: YOLO5S, YOLO5M, YOLO5L and YOLO5X. Figure 3.20 shows a comparison between the versions.
The idea to use images from Twitter is to observe the objects that each model will detect, and choose the one that presents the biggest advantage on salient areas detection. The

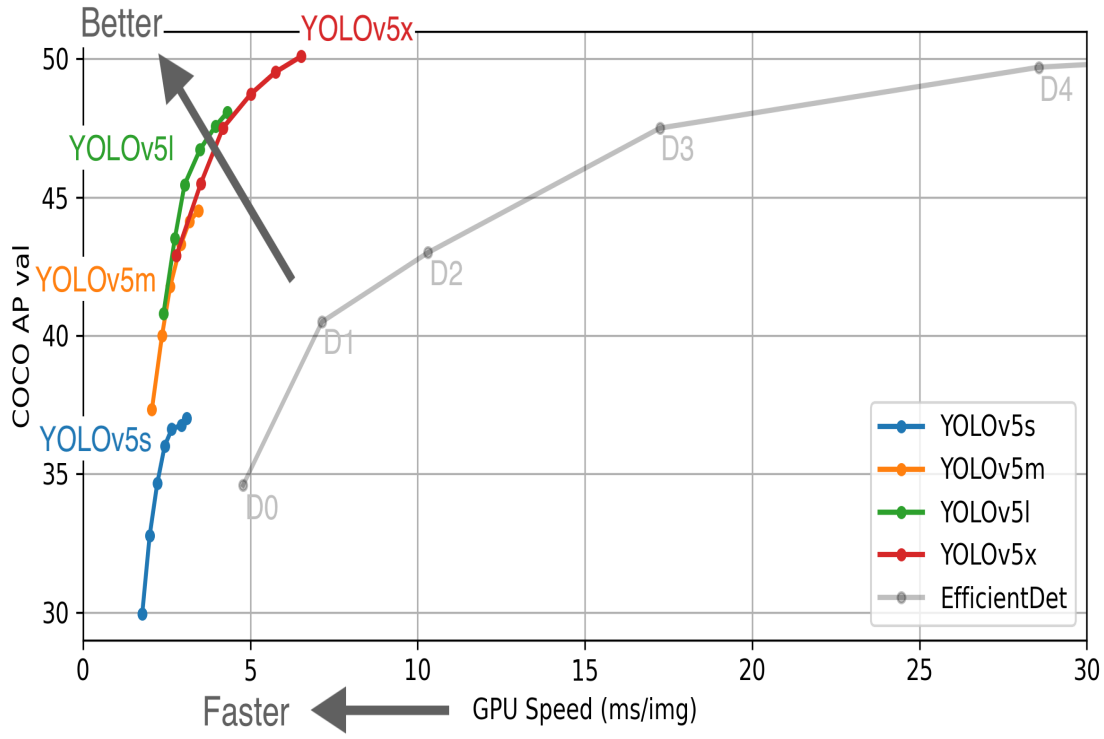


Figure 3.20: Comparison between the detection accuracy and performance of the available models. (Source: image from [Ult]).

reason to try to use an object detector trained with custom data is to take advantage not only to detect salient areas but also to identify the object classes and to use this information.

The MakeSense tool was used to label the objects in the images. 10 classes were selected, which are: State authority, State vehicle, Fire, Smoke, Flag, Poster, Person, Car, Weapon, and Damage.

The tests were made in order to decide the model to be used for the identification and selection of salient areas on images. For the tests, 3 datasets were used:

- **Twitter dataset (custom dataset):** the images collected directly from Twitter;
- **Twitter dataset with data augmentation;**
- **VOC dataset:** one of the most generic and largest datasets.

Each dataset was tested with each sub version of YOLO5, which are: YOLO5X, YOLO5S, YOLO5M, and YOLO5L. All the training was set up with 130 epochs. For Twitter Dataset and Twitter Dataset with augmentation, the batch size used was 512, for the VOC dataset, the batch size was 64.

3.5.1 Twitter Dataset

For the first set of tests, the Twitter Dataset was used, which contains 417 images, and the respective (10) classes. The data was split into: 333 (80%), 42 (10%), and 42 (10%) for train, validation, and test, respectively. Regarding execution time on training, the

YOLO5S model was the fastest (0.127 hours). YOLO5X and YOLO5L, presented very similar results, with a small difference between their mAP values. However, the difference in execution time was significant. Therefore, YOLO5L was faster than YOLO5X.
Figure 3.21 shows the confusion matrix obtained with the YOLO5L model since it has reached the best performance on the tests made.

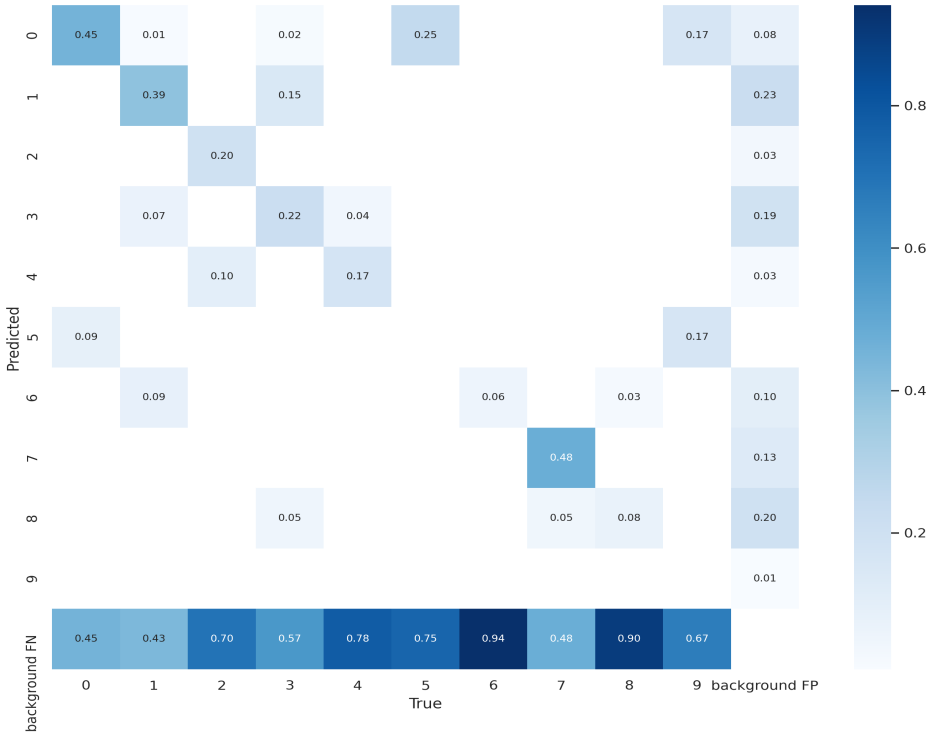


Figure 3.21: Confusion matrix obtained when training the YOLO5L model in the Twitter Dataset.

3.5.2 Twitter Dataset with augmentation

For this set of tests, the Twitter Dataset with augmentation was used. The following augmentation procuderes were applied:

- **Flip:** horizontal, vertical;
- **90° Rotate:** clockwise, counter-clockwise, upside down;
- **Crop:** 0% minimum zoom, 42% maximum zoom;
- **Shear:** ±25° horizontal, ±22° vertical;
- **Blur:** up to 2.25px.

The dataset contains 1041 images and the same 10 classes. The data was split into: 936 (90%), 63 (6%), and 42 (4%) images for train, validation, and test, respectively. Despite the higher number of images, through the tests made, it was possible to see a decay on YOLO's performance, since any result overcame the results obtained on the tests made with the original dataset (Twitter Dataset without augmentation). Figure 3.22 shows the confusion matrix obtained with the model that reached the best results.

Figure 3.22: Confusion matrix obtained when training the YOLO5L model in the Twitter Dataset with augmentation.

3.5.3 VOC Dataset

The VOC dataset contains 21,503 images with annotations. The data was split into: 16,551, and 4,952 images for train and validation, respectively. It was only possible to train using the YOLO5X model. With the other models, even trying a lower batch size, a few epochs were achieved before returning "core dumped". Figure 3.23 shows the confusion matrix obtained from the YOLO5X model.

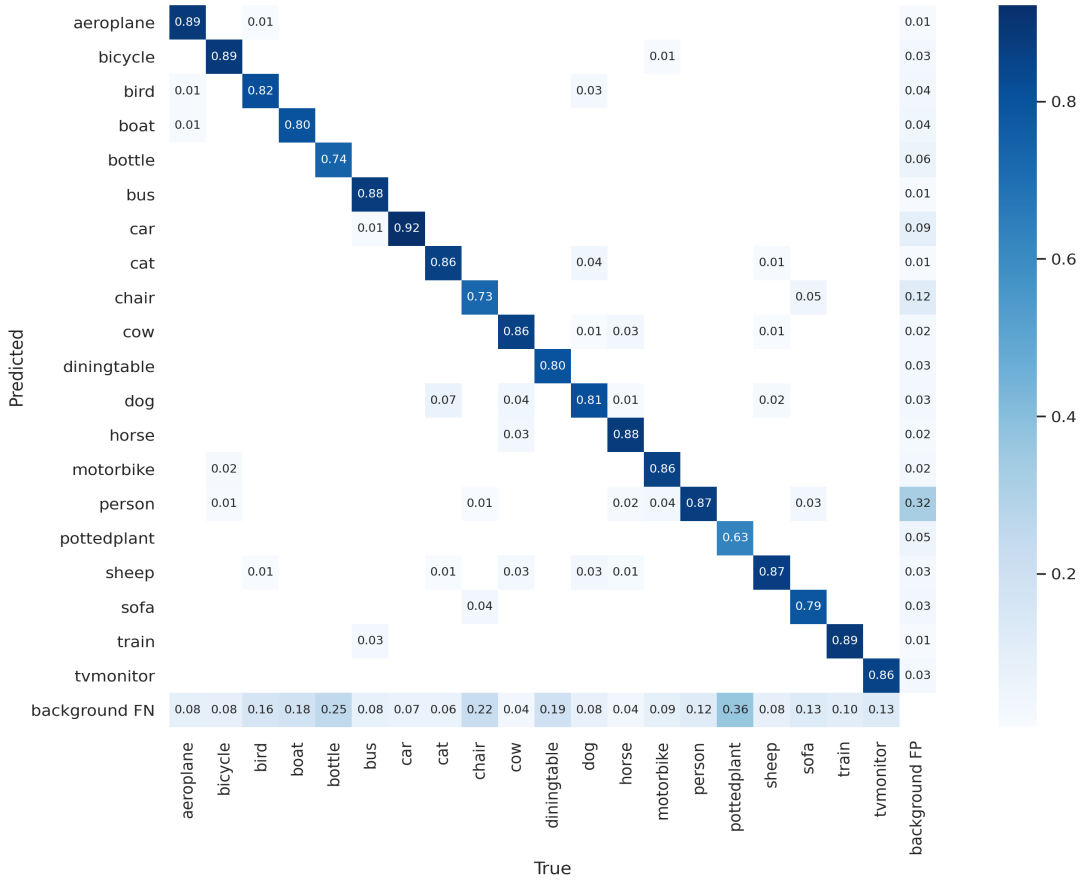


Figure 3.23: Confusion matrix obtained when training the YOLO5X model in the VOC Dataset.

Table 3.3 shows an overview of the results obtained from the tests made.

	Twitter Dataset			**Twitter Dataset_Aug**			**VOC Dataset**		
	mAP.5	mAP.95	Time	mAP.5	mAP.95	Time	mAP.5	mAP.95	Time
YOLO5S	17.9%	6.6%	0.127	13.8%	4.09%	0.313	-	-	-
YOLO5M	16.1%	5.13%	0.230	14.5%	5.00%	0.420	-	-	-
YOLO5L	19.4%	7.3%	0.364	16.0%	5.61%	0.903	-	-	-
YOLO5X	19.7%	7.2%	0.583	13.2%	4.95%	1.511	83.1%	62.7%	27.309

Table 3.3: Comparison between the results obtained from the tests.

Despite the low accuracy obtained by the model trained with the Twitter Dataset, such behavior is comprehensible, because regarding the relation between the data amount in the dataset and the number and type of classes, it is expected a large size dataset, just as we

see the amount of data in the VOC dataset and the amount for the amount of classes that can be recognized. Also, for objects like person and state_authority, it is expected that the model makes confusion between both, mainly when the object is with a low resolution and relatively far from the focus.

However, despite the model trained with the VOC Dataset reaching a higher accuracy (since it is a large dataset), this does not mean it has better utility for this project. Notwithstanding the model trained with the Twitter Dataset achieved a low accuracy (on object recognition), its detected areas could have a bigger significance than the areas detected by the model trained with VOC, since VOC annotation includes: bird, cow, horse, sheep, etc.

Figure 3.24 presents an example of salient areas obtained from an image.

Figure 3.24: Example of using the salient area detector using YOLO model trained with Twitter dataset.

Hence, it is necessary to compare the areas that each model will detect, in order to decide which model can be useful for this project. Therefore, both versions will be tested in the final pipeline in order to understand which version has the greatest advantages in detecting protruding areas.

Despite being images that protrude from the original image, it makes no sense to prepare another image classification model, just to analyze these images. Therefore, it was decided to reuse the global image classifier, but instead of receiving the entire image, it would receive the salient images and finally return the class that was obtained with the highest degree of confidence. This way, consistency in how images are analyzed would be maintained, and time would be saved.

3.6 Text Classification

The sentiment analysis can be made with text. Text sentiment analysis is a procedure derived from Natural Language Processing (NLP). We can have three levels of sentiment analysis [ASM17]:

- **Document level:** the entire document will be considered as a single entity. Thus, the analysis will be applied to the whole document;
- **Sentence-level:** every sentence will be considered as a single entity. Thus, the analysis will be applied to individual sentences. The result will be obtained by calculating the overall result of the document;
- **Aspect-level:** it's fine-grained to discover sentiments on aspects of items. The positive and negative opinion is identified from the already extracted features.

However, this project does not include the development of a text classifier, which will be made by other members of the team. Therefore, for this role, three methods which use different approaches to do sentiment analysis were tested: TextBlob [Lor], Valence Aware Dictionary and sEntiment Reasoner (VADER) [HG15], and the text model proposed in [LGAC21] (described in section 2.7).
TextBlob is a Python (2 and 3) library for processing textual data. It provides a simple API for diving into common NLP tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more. TextBlob will ignore the words that it does not know, it will consider words and phrases that it can assign polarity to and averages to get the final score, it will provide:

- **Polarity:** is a float that lies between [-1,1], where -1 indicates negative sentiment and +1 indicates positive sentiments;
- **Subjectivity:** is also a float that lies in the range of [0,1]. Subjective sentences generally refer to opinion, emotion, or judgment.

VADER is a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media. It does not require to be trained because it is constructed through a standard sentiment lexicon, and it uses a list of lexical features (e.g. word) which are labeled as positive or negative according to their semantic orientation to calculate the text sentiment [HG15]. VADER sentiment returns the probability of a given input sentence to be positive, negative, and neutral.
The main disadvantage with the rule-based approach for sentiment analysis is that the method only considers individual words and completely neglects the context in which it is used. For example, "the party was savage" will be negative when considered by any token-based algorithms.
As there was a lot of information regarding the accuracy of each classifier, it was ideal to perform the tests with both methods.
We know that tweets involve a lot of noise, such as emojis/emoticons, links, numbers, etc. Therefore, before going through one of the text evaluation methods, the respective tweet

will be cleaned, in order to remove this noise and only the clear text to be evaluated.

First, the tweet will be passed by Beautiful Soup, it's a Python library for pulling data out of HTML and eXtensible Markup Language (XML) files. It works with the parser to provide idiomatic ways of navigating, searching, and modifying the parse tree. It will be used to avoid HTML encoding that has not been converted to text, and ended up in the text field as '&','&quot',etc., which means decoding HTML to general text.

The second part of the preparation is dealing with @mention. Even though @mention carries some information (which another user that the tweet mentioned), this information does not add value to build a sentiment analysis model.

The third part is dealing with URL links, wich as, with @mention, even though it carries some information, for sentiment analysis purpose, it will be ignored.

There is the possibility of Unicode Transformation Format (UTF)-8 Byte Order Mark (BOM) character issues. The UTF-8 BOM is an array of bytes (EF BB BF) that allows the reader to recognize a file as being encoded in UTF-8. To avoid unfamiliar characters, we used a text decoder that replaces them by the symbol "?".

Sometimes the text used with **hashtags** can give useful information about the tweet. It might be a bit risky to remove the hashtags. So it was decided to leave the text intact and just remove the symbol (#). It will be employed by cleaning all the non-letter characters (including numbers). Then the text will be transformed to lower case.

During the letters-only process, unnecessary white space is created, so redundant white spaces is removed.

After the tweet is cleaned, the classification method will be called.

The polarity value will be obtained, which will be converted into one of the three classes. Just in case the polarity is neutral (which means TextBlob returned 0 from the classification), the confidence degree is automatically set to 1 (in order to prevent 0 from being returned and negatively influence the final classification of the model).

Unlike TextBlob, VADER returns the degree of probability for each class, so the index (corresponding to each class) of the highest value in the list of degrees of probability will be returned. Then, for VADER, we can actually use its degree of probability as the confidence degree value.

Therefore, from the text model, a tuple will be returned containing the class and the confidence degree.

3.7 Final Model

To make the fusion of the information of each model, we propose three different methods: i) considering the average of all models, ii) using a voting system, and iii) using AutoML. To obtain the average of all models, each class obtained by the model and its respective accuracy will be multiplied, and this value will be divided by the number of models that were evoked, that is, to consider only the information of the models that were actually evoked, which is represented by Equation 3.1.

$$class = \frac{1}{n}\sum_{i=1}^{n} X_i p_i \quad (3.1)$$

We will also consider a variation to the method 1, in which the confidence degree associated to which class will also be used. Thus, the variation of Equation 3.1 will consider the confidence degrees information returned by each model together with the class, which is represented by Equation 3.2,

$$class = \frac{1}{n}\sum_{i=1}^{n} X_i p_i k_i \quad (3.2)$$

where:

- n is the number of used models;
- X_i is the polarity obtained by the models;
- p_i is the accuracy values for the validation set models;
- k_i is the confidence degree value associated to the polarity by the models.

After obtaining its result, it will be sent to a function f in order to return one of the 3 classes, thus obtaining an integer value. This function will receive the value obtained from the average and will return the class:

$$f(class) = \begin{cases} 0, & \text{if } class \leq 0.33 \\ 1, & \text{if } 0.33 < class \leq 0.66 \\ 2, & \text{if } class > 0.66 \end{cases} \quad (3.3)$$

For the second method we decided to use an AutoML approach [HKV19]. AutoML is the process of automating the end-to-end process of applying machine learning to real-world problems. AutoML tends to automate the maximum number of steps with a minimum amount of human effort [HZC21]. We decided to use H2O AutoML, which is a fully open-source, distributed in-memory machine learning platform with linear scalability. H2O

supports the most widely used statistical & machine learning algorithms [LeD20]. We decided to use this approach and take advantage of AutoML for finding the final classifier that works on top of the the decisions of the individual models, for this we will infer the train set with the proposed model, and then the values returned by each model (the class and the respective confidence degree) will be used as input in the AutoML.
For the voting system we will count the votes for each class, and to avoid any tie the accuracy value of each model will be considered, when necessary. Therefore, a tuple will be created, which will store the vote count of each class and the sum of the accuracy values of each model that voted in this same class:

$$(v_i, s_i), \quad i = 0, 1, 2, \quad (3.4)$$

where v_i is the vote count for class i and s_i is the sum of the accuracy values for that class. The selected class is given by:

$$class = \arg\max_i v_i \quad (3.5)$$

when there is no draw between the votes, and in the case of draw, the class is given by:

$$class = \arg\max_i s_i, \quad (3.6)$$

where in this case, the index i runs through the drawn classes only. Then, to obtain the winner class, we will consider the index of the tuple with the highest s_i value. If there is no tie, the tuple with the highest number of votes will be returned. In case of a tie, the tuple (among those that are tied) with the highest sum s_i will be returned.

3.8 Conclusions

This chapter presented each component of the proposed method. From the individual performance of each component, it is possible to have an idea that the proposed method will work as expected. In the next chapter, the experiments will be carried out and the results discussed, in order to observe the model's behavior.

Chapter 4

Results and Discussion

4.1 Introduction

This chapter presents the experiments and a reflection on the results obtained. This chapter is split as follows: section 4.2 presents the proposed tests to be executed and the dataset used for this purpose. The following section 4.3 presents the results and a discussion about the values obtained. Section 4.4 presents a comparison between the values obtained by the proposed method and works in the area that used the same dataset. Finally, section 4.5 contains the main conclusions made while formulating this chapter.

4.2 Experiments and Results

For the experiments, a variation of the B-T4SA validation set was used. This dataset was presented in subsection 2.2.0.10, which is a dataset composed of social media (Twitter) content (images and tweets). B-T4SA is a balanced subset of T4SA, where the corrupted and near-duplicate images have been removed.
The original B-T4SA validation set is composed by 51,000 samples. However, due to the hardware and time limitations, this set had to be randomly decreased approximately 82%, resulting 9,064 samples. Therefore, with the original test set the tests had a duration of approximately between 20-22 hours, and with the reduction the tests were reduced to approximately between 6-9 hours.
In order to evaluate the proposed method and its behaviour, 10 types of tests were stipulated, which are:

- **Test 1:** the proposed method is fully tested, that is, all the models were used during the tests. However, the confidence degree of each model will not be used to calculate the final average;
- **Test 2:** the proposed method is fully tested, that is, all the models were used during the tests. However, the confidence degree of each model will be used to calculate the final average;
- **Test 3:** the proposed method is fully tested, that is, all the models were used during the tests. However, the confidence degree of each model, except for the text and salient areas classifiers, will be used to calculate the final average (this test has three variations);
- **Text 4:** only the global image and text classifiers will be used, that is, without using the proposed models;

- **Test 5:** the model will be tested only using global image, salient areas, and text models;
- **Test 6:** the model will be tested only using global image, text, and facial expression recognition models;
- **Test 7:** only the global image model will be used;
- **Test 8:** only the salient area model will be used;
- **Test 9:** only the text model will be used;
- **Test 10:** only the FER model will be used.

With these experiments, we should expect to obtain the best configuration to then proceed with the test using the respective dataset, which is composed by 51,000 samples. We decided to follow this approach due to the execution time for each test using the entire validation dataset.
The experiments were made in a computer with the following specifications:

- CPU: Advanced Micro Devices (AMD) Ryzen 7 2700 (Octacore | 16 Threads) 3.2GHz;
- Random-Access Memory (RAM): 16GB;
- GPU: NVIDIA 1080ti;
- Disk: 256BG SSD, 3TB Hard Disk Drive (HDD).

4.3 Discussion

4.3.1 Test 1

These tests were made in order to evaluate the proposed method's accuracy without considering the confidence degree of each model on the final average. Table 4.1 presents the results obtained and Table 4.2 presents the respective times. Since Test 1 does not involve the confidence degree, it was possible to fuse the information using the three methods presented in Section 3.7.

	Text Configuration			YOLO Model		Accuracy		
Test	**TextBlob**	**VADER**	**R-CNN**	**Twitter**	**VOC**	**Mean**	**AutoML**	**V.S.**
Test001	X	-	-	X	-	50.60%	-	59.06%
Test002	-	X	-	X	-	48.11%	-	49.59%
Test003	-	-	X	X	-	58.86%	56.29%	72.31%
Test004	X	-	-	-	X	50.72%	-	58.98%
Test005	-	X	-	-	X	48.03%	-	49.10%
Test006	-	-	X	-	X	59.31%	57.52%	**73.19%**

Table 4.1: The results obtained from the tests, which were made in order to evaluate the proposed method's accuracy without considering the confidence degree of each model on the final average.

Test	Fusion Time (H)			Decision Time (H)		
	Mean	AutoML	V.S.	Mean	AutoML	V.S.
Test001	0.83e-03	-	0.12e-02	6.37	-	6.37
Test002	0.83e-03	-	0.12e-02	6.20	-	6.20
Test003	0.83e-03	1.47	0.12e-02	5.52	6.39	5.52
Test004	0.83e-03	-	0.12e-02	9.28	-	9.28
Test005	0.83e-03	-	0.12e-02	9.31	-	9.31
Test006	0.83e-03	1.47	0.12e-02	9.15	10.57	9.15

Table 4.2: The execution time resulted from the tests, which were made in order to evaluate the proposed method's accuracy without considering the confidence degree of each model on the final average.

From the tests made with TextBlob and VADER, it was possible to observe a little variation in the accuracy obtained. The accuracy value keeps between 48%-51%. The setup using the YOLO model trained with the VOC dataset and the VADER classifier (TestE), obtained the lowest accuracy between all the tests, 48.03%. However, the setup using the YOLO model trained with the Twitter dataset and the VADER classifier (TestB), performed better (48.11%). Despite the small difference between the values, it was possible to observe an increase of 1.54%. Using the voting system to fuse the information of all models, we can see an improvement of 3.08% for Test B and 2.23% for Test E on the accuracy obtained.
Almost the same happens with the tests using TextBlob. The setup using the YOLO model trained with the Twitter dataset and the TextBlob classifier (TestA), obtained a low accuracy (50.60%). However, when compared with the lowest value obtained using VADER (48.03% - value obtained from TestE), we can notice an increase of 5.35%. The setup using the YOLO model trained with the VOC dataset and the TextBlob classifier (TestD), performed better (50.72%). Despite the small difference between the values from TestA and TestD, it was possible to observe an increase of 0.24%. When compared with the highest value obtained using VADER (48.11% - value obtained from TestB), we can notice an increase of 5.43%. However, unlike VADER, the voting system for the configurations using TextBlob brought advantages. We can notice a higher difference between the values. We can observe an improvement of 16.72% for Test A and 16.29% for Test D.
Like in [GA19b], TextBlob still presents better performance when compared to VADER.

However, using the R-CNN developed in [LGAC21], described in Section 2.7, which has the purpose of classifying text, we can observe a significant difference when compared with the tests where TextBlob and VADER were used. The setup using the YOLO model trained with the VOC dataset and the R-CNN classifier (TestF), obtained the lowest accuracy (58.86%) from the tests using R-CNN. However, even comparing its result with the highest values obtained with TextBlob (50.72%), and VADER (48.11%) we can notice an increase of 16.05% and 22.34% over the value, respectively. Similarly to the configurations using TextBlob, we can notice a higher difference between the values when using the voting system for the configurations using the R-CNN. We can observe an improvement of 15.47% for TestC and 18.26% for TestF.
Using AutoML, we could notice a drop in the performance of about 4.37% and 22.15% for

TestC when compared with using using the mean approach and voting system, respectively. Also, we could notice a drop in the performance of about 3.02% and 21.41% for TestC when compared with using using the mean approach and voting system, respectively.

It was possible to verify that using the voting system, the model reaches high accuracy values, instead of fusing the information by calculating a mean between the values. This difference is noticeable mainly for configurations that use TextBlob and R-CNN.

The setup using the YOLO model trained with the Twitter dataset and the R-CNN classifier (TestC), obtained the highest accuracy between all the tests, 59.31%. When compared with the highest values obtained with TextBlob (50.72%), and VADER (48.11%) we can notice an increase of 16.94% and 23.28% over the value, respectively. Comparing the tests using R-CNN, we can see an increase of 0.76% between TestC and TestF (using the voting system).

Like B-T4SA, the VOC dataset is composed of a variety of categories. However, the Twitter dataset, used to train the YOLO model from scratch, was created in order to identify negative situations. Thus, there is a high probability that the YOLO model trained from scratch will lose relevant areas, just because there is no negative information to be considered. Thus, it will be normal to see cases where the YOLO model trained with the VOC dataset outperforms the YOLO model trained with the Twitter dataset, and vice versa.

Thus, we can clearly see the difference when using a R-CNN for text classification, instead of using the Python libraries.

4.3.2 Test 2

These tests were made in order to evaluate the proposed method's accuracy considering the confidence degree of each model on the final average. Table 4.3 presents the results obtained and Table 4.4 presents the respective times. Unlike Test 1, the confidence degree will be used for these tests. Therefore, it will not be possible to use the voting system to fuse the information.

	Text Configuration			YOLO Model		Accuracy	
Test	**TextBlob**	**VADER**	**R-CNN**	**Twitter**	**VOC**	**Mean**	**AutoML**
Test007	X	-	-	X	-	47.83%	-
Test008	-	X	-	X	-	47.21%	-
Test009	-	-	X	X	-	70.59%	56.28%
Test010	X	-	-	-	X	47.76%	-
Test011	-	X	-	-	X	46.79%	-
Test012	-	-	X	-	X	**71.13%**	57.13%

Table 4.3: The results obtained from the tests, which were made in order to evaluate the proposed method's accuracy considering the confidence degree of each model on the final average.

From the tests made with TextBlob and VADER, it was possible to observe a little variation in the accuracy obtained. The accuracy value keeps between 46%-48%. The setup using the YOLO model trained with the VOC dataset and the VADER classifier (TestE), obtained

Test	Fusion Time (H)			Decision Time (H)		
	Mean	AutoML	V.S.	Mean	AutoML	V.S.
Test007	0.12e-02	-	0.14e-02	6.22	-	6.22
Test008	0.12e-02	-	0.14e-02	6.24	-	6.24
Test009	0.12e-02	1.47	0.14e-02	5.50	7.37	5.50
Test010	0.12e-02	-	0.14e-02	9.36	-	9.36
Test011	0.12e-02	-	0.14e-02	9.29	-	9.29
Test012	0.12e-02	1.47	0.14e-02	9.10	10.57	9.10

Table 4.4: The execution time resulted from the tests, which were made in order to evaluate the proposed method's accuracy considering the confidence degree of each model on the final average.

the lowest accuracy between all the tests, 46.79%. However, the setup using the YOLO model trained with the Twitter dataset and the VADER classifier (TestB), performed better (47.21%). Despite the small difference between the values, it was possible to observe an increase of 0.90%.

The same happens with the tests using TextBlob. The setup using the YOLO model trained with the VOC dataset and the TextBlob classifier (TestD), obtained a low accuracy (47.76%). However, when compared with the lowest value obtained using VADER (46.79% - value obtained from TestE), we can notice an increase of 2.07%. The setup using the YOLO model trained with the Twitter dataset and the TextBlob classifier (TestA), performed better (47.83%). Despite the small difference between the values from TestA and TestE, it was possible to observe an increase of 0.15%. When compared with the highest value obtained using VADER (47.21% - value obtained from TestB), we can notice a small increase of 1.31%.

However, using the R-CNN we can observe a significant difference. The setup using the YOLO model trained with the Twitter dataset and the R-CNN classifier (TestC), obtained the lowest accuracy (70.59%) from the tests using R-CNN. However, even comparing its result with the highest values obtained with TextBlob (47.83%), and VADER (47.76%) we can notice a significant increase of 47.59% and 47.80% over the value, respectively. The setup using the YOLO model trained with the VOC dataset and the R-CNN classifier (TestF), obtained the highest accuracy between all the tests, 71.13%. When compared with the highest values obtained with TextBlob (47.83%), and VADER (47.76%) we can notice an increase of 48.71% and 48.93% over the value, respectively. Comparing the tests using R-CNN, we can see an increase of 0.76% between TestC and TestF.

Using AutoML, we could notice a drop in the performance of about 20.27% for TestC when compared with using using the mean approach. Also, we could notice a drop in the performance of about 20.88%, for TestF.

Comparing with the results obtained on subsection 4.3.1, the proposed method's performance dropped approximately 5.70%, and 1.87% when using TextBlob and VADER, respectively. However, the opposite happens then using the R-CNN model, the proposed method's performance has an increase of 20.3%. Therefore, we could notice using the confidence degree of each model negatively influenced the proposed method's performance when using TextBlob and VADER. However, the confidence degree positively influenced

the proposed method's performance when using the R-CNN model, which means the results from the R-CNN are obtained with a higher accuracy than the results obtained with TextBlob and VADER, what makes the R-CNN model more reliable.
Also, we could notice the difference between the YOLO models. However, despite the small difference between the accuracy values, when observing the execution time we can see a significant difference. The tests using the YOLO model trained with the Twitter dataset finished almost 3 hours earlier than the tests using the YOLO model trained with the VOC dataset, and considering the difference obtained between TestC and TestF, it does not seem to be so advantageous to wait 3 hours to improve the accuracy value by 0.76%.

4.3.3 Test 3

Considering the behaviour observed on subsection 4.3.2, intuitively, we thought of disregarding the degrees of confidence of the variable model (between the tests), that is, of the salient areas (YOLO trained from scratch and YOLO trained with VOC dataset) and text classifiers (TextBlob and VADER). Therefore, Test 3 has three variations, which are: **i)** the proposed method is fully tested and the confidence degree of each model, except for the text classifier, will be used to calculate the final average; **ii)** the proposed method is fully tested and the confidence degree of each model, except for the salient areas classifier, will be used to calculate the final average; **iii)** the proposed method is fully tested and the confidence degree of each model, except for the text and salient areas classifiers, will be used to calculate the final average.
Table 4.5 presents the results obtained and and Table 4.6 presents the respective times.

4.3.3.1 Results without using text classification confidence degree

Analysing the set of experiments where the text confidence degree was not considered, from the tests made with TextBlob and VADER, it was possible to observe a variation in the accuracy obtained. The accuracy value keeps between 47%-54%. The setup using the YOLO model trained with the Twitter dataset and the VADER classifier (TestB), obtained the lowest accuracy between all the tests, 47.62%. However, the setup using the YOLO model trained with the VOC dataset and the VADER classifier (TestE), performed better (48.12%). Despite the small difference between the values, it was possible to observe an increase of 1.05%.
The same happens with the tests using TextBlob. The setup using the YOLO model trained with the Twitter dataset and the TextBlob classifier (TestA), obtained a low accuracy (52.32%). However, when compared with the lowest value obtained using VADER (47.62% - value obtained from TestB), we can notice an increase of 9.87%. The setup using the YOLO model trained with the VOC dataset and the TextBlob classifier (TestD), performed better (53.28%). Despite the small difference between the values from TestA and TestD, it was possible to observe an increase of 1.83%. When compared with the highest value obtained using VADER (48.12% - value obtained from TestE), we can notice an increase of 10.72%. However, using the R-CNN we can observe a significant difference. The setup using the

	Text Configuration			YOLO Model		Conf. Degree		Accuracy		Time(H)	
Test	**TextBlob**	**VADER**	**R-CNN**	**Twitter**	**VOC**	**Text**	**SA**	**Mean**	**AutoML**		
Test013	X	-	-	X	-	-	X	52.32%	-	6.05	-
Test014	-	X	-	X	-	-	X	47.62%	-	6.10	-
Test015	-	-	X	X	-	-	X	70.51%	56.28%	5.52	1.47
Test016	X	-	-	-	X	-	X	53.28%	-	9.37	-
Test017	-	X	-	-	X	-	X	48.12%	-	11.14	-
Test018	-	-	X	-	X	-	X	**71.33%**	57.13%	9.11	1.47
Test019	X	-	-	X	-	X	-	48.67%	-	6.25	-
Test020	-	X	-	X	-	X	-	48.21%	-	6.30	-
Test021	-	-	X	X	-	X	-	68.54%	56.43%	5.53	1.47
Test022	X	-	-	-	X	X	-	48.95%	-	9.27	-
Test023	-	X	-	-	X	X	-	47.80%	-	9.31	-
Test024	-	-	X	-	X	X	-	**69.09%**	58.24%	9.24	1.47
Test025	X	-	-	X	-	-	-	52.16%	-	6.27	-
Test026	-	X	-	X	-	-	-	47.99%	-	6.30	-
Test027	-	-	X	X	-	-	-	68.55%	56.26%	6.15	1.47
Test028	X	-	-	-	X	-	-	52.49%	-	9.24	-
Test029	-	X	-	-	X	-	-	48.85%	-	9.25	-
Test030	-	-	X	-	X	-	-	**68.89%**	57.43%	9.05	1.47

Table 4.5: The results obtained from the tests, which were made in order to evaluate the proposed method's accuracy considering the confidence degree of each model, except for the salient areas (SA) and text classifiers, on the final average

	Fusion Time (H)			Decision Time (H)		
Test	**Mean**	**AutoML**	**V.S.**	**Mean**	**AutoML**	**V.S.**
Test013	0.97e-03	-	0.11e-02	6.05	-	6.05
Test014	0.97e-03	-	0.11e-02	6.10	-	6.10
Test015	0.97e-03	1.47	0.11e-02	5.52	7.39	5.52
Test016	0.97e-03	-	0.11e-02	9.37	-	9.37
Test017	0.97e-03	-	0.11e-02	9.45	-	9.45
Test018	0.97e-03	1.47	0.11e-02	9.11	10.58	9.11
Test019	0.93e-03	-	0.98e-03	6.25	-	6.25
Test020	0.93e-03	-	0.98e-03	6.30	-	6.30
Test021	0.93e-03	1.47	0.98e-03	5.53	7	5.53
Test022	0.93e-03	-	0.98e-03	9.27	-	9.27
Test023	0.93e-03	-	0.98e-03	9.31	-	9.31
Test024	0.93e-03	1.47	0.98e-03	9.24	11.11	9.24
Test025	0.90e-03	-	0.96e-03	6.27	-	6.27
Test026	0.90e-03	-	0.96e-03	6.30	-	6.30
Test027	0.90e-03	1.47	0.96e-03	6.15	7.12	6.15
Test028	0.90e-03	-	0.96e-03	9.24	-	9.24
Test029	0.90e-03	-	0.96e-03	9.25	-	9.25
Test030	0.90e-03	1.47	0.96e-03	9.05	10.52	9.05

Table 4.6: The execution time resulted from the tests, which were made in order to evaluate the proposed method's accuracy considering the confidence degree of each model, except for the salient areas (SA) and text classifiers, on the final average

YOLO model trained with the Twitter dataset and the R-CNN classifier (TestC), obtained the lowest accuracy (70.51%) from the tests using R-CNN. However, even comparing its result with the highest values obtained with TextBlob (53.28%), and VADER (48.12%) we can notice a significant increase of 32.34% and 46.53% over the value, respectively. The setup using the YOLO model trained with the VOC dataset and the R-CNN classifier (TestF), obtained the highest accuracy between all the tests (not using the text classifier confidence degree), 71.33%. When compared with the highest values obtained with TextBlob (53.28%), and VADER (48.12%) we can notice an increase of 33.88% and 48.23% over the value, respectively. Comparing the tests using R-CNN, we can see an increase of 1.16% between TestC and TestF.
Using AutoML, we could notice a drop in the performance of about 20.18% for TestC when compared with using using the mean approach. Also, we could notice a drop in the performance of about 19.91%, for TestF.
Comparing the highest accuracy obtained in the subsection 4.3.2, using TextBlob and VADER, it was possible to observe an increase of approximately 11.39%, and 1.80% on the accuracy obtained, respectively. Therefore, it was possible to conclude that the confidence degree values can positively influence the model. However, it is necessary to exclude the confidence degree of the text classifier when using TextBlob and VADER.
Analysing the results where the R-CNN was used, comparing the highest accuracy obtained in the subsection 4.3.2, it was possible to observe a small increase of approximately 0.28% on the accuracy obtained.

4.3.3.2 Results without using salient area confidence degree

Analysing the set of experiments where the salient areas confidence degree was not considered, from the tests made with TextBlob and VADER, it was possible to observe a little variation in the accuracy obtained. The accuracy value keeps between 47%-49%. The setup using the YOLO model trained with the VOC dataset and the VADER classifier (TestK), obtained the lowest accuracy between all the tests, 47.80%. However, the setup using the YOLO model trained with the Twitter dataset and the VADER classifier (TestH), performed better (48.21%). Despite the small difference between the values, it was possible to observe an increase of 0.86%.
When analysing the tests using TextBlob, the setup using the YOLO model trained with the Twitter dataset and the TextBlob classifier (TestG), obtained a low accuracy (48.67%). However, when compared with the lowest value obtained using VADER (47.80% - value obtained from TestK), we can notice an increase of 1.82%. The setup using the YOLO model trained with the VOC dataset and the TextBlob classifier (TestJ), performed better (48.95%). Despite the small difference between the values from TestG and TestJ, it was possible to observe an increase of 2.69%. When compared with the highest value obtained using VADER (48.21% - value obtained from TestH), we can notice an increase of 1.53%.
The setup using the YOLO model trained with the Twitter dataset and the R-CNN classifier (TestI), obtained the lowest accuracy (68.54%) from the tests using R-CNN. However, even comparing its result with the highest values obtained with TextBlob (48.95%),

and VADER (48.21%) we can notice a significant increase of 40.02% and 42.17% over the value, respectively. The setup using the YOLO model trained with the VOC dataset and the R-CNN classifier (TestL), obtained the highest accuracy between all the tests, 69.09%. When compared with the highest values obtained with TextBlob (48.95%), and VADER (48.21%) we can notice an increase of 41.14% and 43.31% over the value, respectively. Comparing the tests using R-CNN, we can see an increase of 0.80% between TestI and TestL.
Using AutoML, we could notice a drop in the performance of about 17.67% for TestC when compared with using using the mean approach. Also, we could notice a drop in the performance of about 15.70%, for TestF.
Comparing the highest accuracy obtained in the subsection 4.3.2, using TextBlob and VADER, it was possible to observe an increase of approximately 2.34%, and 2.12% on the accuracy obtained, respectively. Analysing the results where the R-CNN was used, comparing the highest accuracy obtained in the subsection 4.3.2, it was possible to observe a drop of approximately 2.87% on the accuracy obtained.
When comparing with the highest values obtained, for each text configuration, in Subsection 4.3.3.1, the accuracy dropped approximately 8.13% using TextBlob. However, using VADER the accuracy increased approximately 0.19%. When comparing with the highest value obtained with the R-CNN model, the accuracy dropped approximately 3.14%.
Therefore, it was possible to conclude that the text confidence degree is more reliable than the salient area confidence degree.

4.3.3.3 Results without using text and salient area confidence degrees

Analysing the set of experiments where the text and salient areas confidence degree were not considered, from the tests made with TextBlob and VADER, it was possible to observe a variation in the accuracy obtained. The accuracy value keeps between 49%-52%. The setup using the YOLO model trained with the Twitter dataset and the VADER classifier (TestN), obtained the lowest accuracy between all the tests, 47.99%. However, the setup using the YOLO model trained with the VOC dataset and the VADER classifier (TestQ), performed better (48.85%). Despite the small difference between the values, it was possible to observe an increase of 1.79%.
We can observe the same behaviour on the tests using TextBlob, the setup using the YOLO model trained with the Twitter dataset and the TextBlob classifier (TestM), obtained a low accuracy (52.16%). However, when compared with the lowest value obtained using VADER (47.99% - value obtained from TestN), we can notice an increase of 8.69%. The setup using the YOLO model trained with the VOC dataset and the TextBlob classifier (TestP), performed better (52.49%). Despite the small difference between the values from TestM and TestP, it was possible to observe an increase of 0.63%. When compared with the highest value obtained using VADER (48.85% - value obtained from TestQ), we can notice an increase of 7.45%.
The setup using the YOLO model trained with the Twitter dataset and the R-CNN classifier (TestO), obtained the lowest accuracy (68.55%) from the tests using R-CNN. How-

ever, even comparing its result with the highest values obtained with TextBlob (52.49%), and VADER (48.85%) we can notice a significant increase of 30.60% and 40.33% when comparing the value obtained with R-CNN (68.55%), TextBlob (52.49%), and VADER (48.85%), respectively. The setup using the YOLO model trained with the VOC dataset and the R-CNN classifier (TestR), obtained the highest accuracy between all the tests, 68.89%. When compared with the highest values obtained with TextBlob (52.49%), and VADER (48.85%) we can notice an increase of 31.24% and 41.02% over the value, respectively. Comparing the tests using R-CNN, we can see an increase of 0.50% between TestO and TestR.

Using AutoML, we could notice a drop in the performance of about 17.93% for TestC when compared with using using the mean approach. Also, we could notice a drop in the performance of about 16.64%, for TestF.

Comparing the highest accuracy obtained in the subsection 4.3.2, using TextBlob and VADER, it was possible to observe an increase of approximately 9.74%, and 3.47% on the accuracy obtained, respectively. Analysing the results where the R-CNN was used, comparing the highest accuracy obtained in the subsection 4.3.2, it was possible to observe a drop of approximately 3.14% on the accuracy obtained.

When comparing with the highest values obtained, for each text configuration, in Subsection 4.3.3.1, the accuracy dropped approximately 1.48% using TextBlob. However, using VADER the accuracy increased approximately 1.52%. When comparing with the highest value obtained with the R-CNN model, the accuracy dropped approximately 3.42%.

However, when comparing with the highest values obtained, for each text configuration, in Subsection 4.3.3.2, the accuracy increases approximately 7.23% using TextBlob. The same happens using VADER, the accuracy increases approximately 1.33%. When comparing with the highest value obtained with the R-CNN model, the accuracy dropped approximately 0.29%.

Thus, between all the experiments made with the configuration of Test 3 and its variations, it was possible to identify that the setup using the R-CNN with the YOLO model trained with the VOC dataset, presented the best performance. This can mean that the confidence degree returned by the text model is more reliable than the confidence degree returned by the salient areas model. However, it was not good enough to overcome the result obtained in subsection 1, where we obtained 73.19% fusing the information with the voting system. Thus, we can observe a drop of approximately 2.54% when comparing them.

4.3.4 Test 4

These tests were made in order to evaluate the model's accuracy without using the proposed methods. This can be considered one of the most important tests that were made because from the results it was possible to conclude if the proposed method is a good approach or not. This test has a peculiarity. Because it only uses 2 models, it is not fair to use the voting system, as it will always be a tie and the model with greater accuracy will always win the vote, which is the text model. This means that the vote is not a very fair one.

Table 4.7 presents the results from this type of test and and Table 4.8 presents the respective times.

Test	Text Configuration			YOLO Model		Accuracy
	TextBlob	VADER	R-CNN	Twitter	VOC	
Test031	X	-	-	-	-	49.86%
Test032	-	X	-	-	-	47.87%
Test033	-	-	X	-	-	**60.22%**

Table 4.7: The results obtained from the tests, which were made using only global image and text classifiers, that is, without using the proposed models.

Test	Fusion Time (H)			Decision Time (H)		
	Mean	AutoML	V.S.	Mean	AutoML	V.S.
Test031	0.83e-03	-	0.97e-03	0.30	-	0.30
Test032	0.83e-03	-	0.97e-03	0.31	-	0.31
Test033	0.83e-03	1.47	0.97e-03	0.30	2.07	0.30

Table 4.8: The execution time resulted from the tests, which were made using only global image and text classifiers, that is, without using the proposed models.

The highest accuracy value obtained was approximately 59.77% using the R-CNN for text classification.
Comparing with the highest accuracy obtained (71.33%), the model presents a drop of approximately 15.58%. Therefore, it is possible to assert the advantage of using the FER and salient areas models, proving the efficiency of using the proposed method.

4.3.5 Test 5

These tests were made in order to evaluate the method's behaviour without one of the proposed sub-models. Therefore, for this type of test, only global image, salient area, and text classifiers were considered. Similar to the experiments presented in Subsection 4.3.1, the experiments made in Test 5 do not involve the confidence degree, so it was possible to fuse the information using the three methods presented in Section 3.7.
Table 4.9 presents the results obtained and and Table 4.10 presents the respective times. We can identify in the tests where VADER was used that it presented the best values. The difference between the YOLO models used was approximately 1.12%. Otherwise, with TextBlob, the difference between the YOLO models used was significant, approximately 8.70%.
From the tests made with TextBlob and VADER, it was possible to observe a little variation in the accuracy obtained. The accuracy value keeps between 50%-52%. The setup using the YOLO model trained with the VOC dataset and the VADER classifier (TestE), obtained the lowest accuracy between all the tests, 50.01%. However, the setup using the YOLO model trained with the Twitter dataset and the VADER classifier (TestB), performed better (50.33%). Despite the small difference between the values, it was possible to observe an

Test	Text Configuration			YOLO Model		Accuracy			Time(H)		
	TextBlob	VADER	R-CNN	Twitter	VOC	Mean	AutoML	V.S.			
Test034	X	-	-	X	-	51.74%	-	59.91%	6.10	-	6.10
Test035	-	X	-	X	-	50.33%	-	58.40%	5.79	-	5.79
Test036	-	-	X	X	-	62.78%	56.29%	72.67%	5.64	1.47	5.64
Test037	X	-	-	-	X	50.77%	-	58.96%	9.14	-	9.14
Test038	-	X	-	-	X	50.01%	-	58.03%	9.24	-	9.24
Test039	-	-	X	-	X	62.25%	57.52%	**72.74%**	8.97	1.47	8.97

Table 4.9: The results obtained from the tests, which were made using only global image, salient areas, and text models.

Test	Fusion Time (H)			Decision Time (H)		
	Mean	AutoML	V.S.	Mean	AutoML	V.S.
Test034	0.1e-02	-	0.12e-02	6.10	-	6.10
Test035	0.1e-02	-	0.12e-02	5.79	-	5.79
Test036	0.1e-02	1.47	0.12e-02	5.64	7.11	5.64
Test037	0.1e-02	-	0.12e-02	9.14	-	9.14
Test038	0.1e-02	-	0.12e-02	9.24	-	9.24
Test039	0.1e-02	1.47	0.12e-02	8.97	10.44	8.97

Table 4.10: The execution time resulted from the tests, which were made using only global image, salient areas, and text models.

increase of 0.64%. Using the voting system to fuse the information of all models, we can see a significant improvement of 16.03% for Test B and 16.04% for Test E on the accuracy obtained.

The same happens with the tests using TextBlob. The setup using the YOLO model trained with the VOC dataset and the TextBlob classifier (TestD), obtained a low accuracy (50.77%). However, when compared with the lowest value obtained using VADER (50.01% - value obtained from TestE), we can notice an increase of 1.52%. The setup using the YOLO model trained with the Twitter dataset and the TextBlob classifier (TestA), performed better (51.74%). Despite the small difference between the values from TestA and TestE, it was possible to observe an increase of 1.91%. When compared with the highest value obtained using VADER (50.33% - value obtained from TestB), we can notice an increase of 2.80%. The voting system for the configurations using TextBlob brought advantages. We can notice a higher difference between the values. We can observe an improvement of 15.79% for Test A and 16.13% for Test D.

However, using the R-CNN we can observe a significant difference. The setup using the YOLO model trained with the VOC dataset and the R-CNN classifier (TestF), obtained the lowest accuracy (62.25%) from the tests using R-CNN. However, even comparing its result with the highest values obtained with TextBlob (51.74%), and VADER (50.33%) we can notice a significant increase of 20.31% and 23.68% over the value, respectively. The setup using the YOLO model trained with the Twitter dataset and the R-CNN classifier (TestC), obtained the highest accuracy between all the tests, 62.78%. When compared with the highest values obtained with TextBlob (51.74%), and VADER (50.33%) we can

notice an increase of 21.34% and 24.74% over the value, respectively. Comparing the tests using R-CNN, we can see an increase of 0.85% between TestF and TestC. Like the configurations using TextBlob and VADER, we notice a higher difference between the values when using the voting system for the configurations using the R-CNN. We can observe an improvement of 15.75% for TestC and 16.85% for TestF.
Using AutoML, we could notice a drop in the performance of about 10.34% and 22.54% for TestC when compared with using using the mean approach and voting system, respectively. Also, we could notice a drop in the performance of about 7.60% and 20.92% for TestC when compared with using using the mean approach and voting system, respectively.
The model holds a good accuracy (72.74%). However, there is a drop in approximately 0.61% in comparison with the highest accuracy (73.19%) obtained. The consistency observed in subsection 4.3.4 was maintained. We can observe that, to achieve a good accuracy, the proposed method needs to be executed with all the proposed models.
Like with the experiments presented in Subsection 4.3.1, it was possible to verify that using the voting system the model reaches higher accuracy values than using the other 2 approaches.

4.3.6 Test 6

Likewise with the tests presented in subsection 4.3.5, the tests presented in this subsection were made in order to evaluate the method's behaviour without one of the proposed sub-models. However, for Test 6, only global image, text, and facial expression classifiers were considered. Table 4.11 presents the results obtained and and Table 4.12 presents the respective times. Similarly to the experiments presented in Subsection 4.3.5, the experiments made in Test 6 do not involve the confidence degree, so it was possible to fuse the information using the three methods presented in Section 3.7.

	Text Configuration			YOLO Model		Accuracy			Time(H)		
Test	**TextBlob**	**VADER**	**R-CNN**	**Twitter**	**VOC**	**Mean**	**AutoML**	**V.S.**			
Test040	X	-	-	-	-	51.93%	-	65.06%	0.49	-	0.49
Test041	-	X	-	-	-	49.34%	-	59.87%	0.50	-	0.50
Test042	-	-	X	-	-	62.63%	56.29%	**82.90%**	0.51	1.47	0.51

Table 4.11: The results obtained from the tests, which were made using only global image, text, and FER models.

Similarly to the results obtained in subsection 4.3.5, the model holds a good accuracy, and we can identify in the test where R-CNN was used that it presented the best value (62.63%). We can observe an increase of approximately 20.60% and 26.94% over the accuracy values obtained by the tests using TextBlob and VADER, respectively.
With the voting system, we can observe a significant increase in the accuracy values, mainly when using the R-CNN. Comparing both results using the R-CNN, we can see an increase of approximately 32.36% when comparing with the result obtained using the mean, and 47.27% when comparing with the result obtained using AutoML. With TextBlob

Test	Fusion Time (H)			Decision Time (H)		
	Mean	AutoML	V.S.	Mean	AutoML	V.S.
Test040	0.1e-02	-	0.12e-02	0.49	-	0.49
Test041	0.1e-02	-	0.12e-02	0.50	-	0.50
Test042	0.1e-02	1.47	0.12e-02	0.51	2.38	0.51

Table 4.12: The execution time resulted from the tests, which were made using only global image, text, and FER models.

and VADER, we can observe an increase of approximately 25.29% and 21.34%, respectively, when comparing the mean and voting system methods to fuse the information.
Unlike the behavior observed in Subsection 4.3.5, where we can see the highest accuracy (72.74%) did not overcome the 73.19% obtained in Subsection 4.3.1. For this experiment we can observe that the accuracy obtained overcame both values in approximately 13.97% and 13.27%, respectively. This is understandable, considering that the global image model and the salient areas model have very similar behavior, since they share the same architecture. However, when we removed the FER model for the experiments presented in Subsection 4.3.5, it was possible to see a drop in the accuracy when comparing with the whole purposed method. However, for these experiments, when we removed the salient areas model, the accuracy presented a significant increase, which led us to conclude that the salient areas might not be as advantageous as we thought.
Using AutoML, we could notice a drop in the performance of about 10.12% and 32.10% for TestC when compared with using using the mean approach and voting system, respectively.
It's possible to observe that the model can handle well with all the negative and positive classes. However, there's a difficulty when predicting neutral instances. The model presented a large rate of positive rating, the same problem is verified while testing FER model, presented in Subsection 4.3.10. Reinforcing these classes could help improve the model's behavior.

4.3.7 Test 7

The Test043 was made in order to evaluate the global image model's behaviour. It was possible to obtain 49.52% of accuracy, which is a good value. However, comparing with the highest accuracy (82.90%), we can observe a drop of approximately 40.27%.

4.3.8 Test 8

These tests were made in order to evaluate the salient area model's behaviour. However, two tests were addressed since there are two versions of YOLO model, which were described in section 3.5. Also, only images with salient areas were considered for this test.
Table 4.13 presents the results obtained.
It's possible to observe that using YOLO model trained with VOC dataset presented the highest accuracy, 47.12%. However, analysing the execution time, it's discouraging to wait

Test	Text Configuration			YOLO Model		Accuracy	Time(H)
	TextBlob	VADER	R-CNN	Twitter	VOC		
Test044	-	-	-	X	-	45.15%	3.34
Test045	-	-	-	-	X	**47.12%**	8.05

Table 4.13: The results obtained from the tests, which were made using only salient areas model.

5 hours just to get an improvement of approximately 4.36%.
Despite the differences, the model presented a similar behaviour. Thus, it was possible to observe experiments where the YOLO model trained with the Twitter dataset presented high accuracy values over the experiments where the YOLO model trained with the VOC dataset was used, and vice versa.

4.3.9 Test 9

These tests were made in order to evaluate the text model's behaviour. Table 4.14 presents the results obtained.

Test	Text Configuration			YOLO Model		Accuracy	Time(H)
	TextBlob	VADER	R-CNN	Twitter	VOC		
Test046	X	-	-	-	-	64.23%	0.0019
Test047	-	X	-	-	-	52.21%	1.02
Test048	-	-	X	-	-	**87.65%**	0.0015

Table 4.14: The results obtained from the tests, which were made using only text model.

It was possible to observe that the R-CNN obtained the highest accuracy. Overcoming approximately 36.46% and 67.88% the accuracy obtained by TextBlob and VADER, respectively.
The high accuracy values, obtained by the tests using only text classifier, may have a bearing on how the data was tagged. The B-T4SA was tagged using AMT, however, there is no certainty that it were humans who tagged the data.

4.3.10 Test 10

The Test049 was made in order to evaluate the FER model behaviour. Differently from the other tests, this was set up to consider only the images that contain faces. From the test, it was possible to obtain 39.84% accuracy with FER model, the test had an execution time of approximately 27 minutes. Only 3,170 images contain faces.

4.4 Effectiveness of the proposed method

Table 4.15 presents an overview of the results obtained from all the tests made.

Setup	Model				Conf. Dregree				Text Conf.	YOLO Model	Accuracy
	GI	SA	Text	FER	GI	SA	Text	FER			
1 (4.3.1)	X	X	X	X	-	-	-	-	R-CNN	VOC	**73.19%**
2 (4.3.2)	X	X	X	X	X	X	X	X	R-CNN	VOC	71.13%
3 (4.3.3)	X	X	X	X	X	X	-	X	R-CNN	VOC	71.33%
4 (4.3.4)	X	-	X	-	-	-	-	-	R-CNN	-	60.22%
5 (4.3.5)	X	X	X	-	-	-	-	-	R-CNN	VOC	72.74%
6 (4.3.6)	X	-	X	X	-	-	-	-	R-CNN	-	**82.90%**

Table 4.15: Overview of the best results obtained from each test.

From Table 4.15, we could observe two configurations that outperformed during the experiments. These configurations were selected to perform the final test using the respective dataset, which is presented in Table 4.16 and and Table 4.17 presents the respective times.

Test	Text Configuration			YOLO Model		Accuracy		Time(H)
	TextBlob	VADER	R-CNN	Twitter	VOC	Mean	Voting System	
Test050	-	-	X	-	-	64.00%	**80.86%**	5.15
Test051	-	-	X	-	X	63.52%	72.77%	34.19

Table 4.16: The results obtained from the final test using the best configurations observed during the validation phase and the test set.

Test	Fusion Time (Sec.)		Decision Time (H)	
	Mean	V.S.	Mean	V.S.
Test050	15	18	5.15	5.18
Test051	10	13	34.19	34.23

Table 4.17: The execution time resulted from from the final test using the best configurations observed during the validation phase and the test set.

Despite the accuracy obtained with the proposed method (using all the models), the effectiveness of using all models together was proven, so that the proposed method outperformed the works that use the B-T4SA dataset, [GA19b, VCC$^+$17b].
Table 4.18 presents an overview of the accuracy values obtained for work that used the B-T4SA dataset.
We can observe that the proposed method overcomes 41.85% the state-of-the-art [VCC$^+$17b] and 39.03% the work proposed in [GA19b]. However, using the voting system to fuse the information, we can increase these values up to 61.60 % and 58.39%, respectively.
However, it was not able to overcome the model proposed in [LGAC21], probably the difference lays on how the fusion of the information was made. Also their model only considers text and images, which results in less noise during the fusing of the information.

Work	Accuracy
VGG-T4SA FT-A [VCC$^+$17b]	51.30%
VGG-T4SA FT-F [VCC$^+$17b]	50.60%
Hybrid-T4SA FT-A [VCC$^+$17b]	49.10%
Hybrid-T4SA FT-F [VCC$^+$17b]	49.90%
Random Classifier [VCC$^+$17b]	33.30%
Multimodal Approach [GA19b]	52.34%
Multimodal Approach [LGAC21]	**95.19%**
Ours$_{Setup1}$	72.77%
Ours$_{Setup6}$	**80.86%**

Table 4.18: Comparison between the results obtained from the models that used the B-T4SA dataset. Where Setup1 means the use of all models during the test, and Setup6 means the use of the Global Image, Text and FER models during the test.

Also, we had a concern noticing that the labeled data in B-T4SA aren't so reliable. During the tests we could observe situations where the label assigned to the instance was not correct. For example the instance presented in Figure 4.1, which has the tweet "I hate smiling ????????" was labeled as negative.

Figure 4.1: During the tests we could observe situations similar to this. When observing the image, mainly the facial expression, and the following text, we can clearly identify irony in the tweet.

In order to demonstrate the model's behavior, the sample presented in Figure 4.1 was classified. Table 4.19 presents the results obtained from the classification of each model. It is possible to observe that the FER model classifies the image as positive with a high value in the degree of confidence. The negative classification obtained by the global image model is understandable since this model was trained with the B-T4SA dataset and

Model	Classification	Confidence Degree
Global Image	Negative	55.22%
Salient Areas	-	-
R-CNN	Negative	99.00%
FER	Positive	96.77%

Table 4.19: The results obtained from the classification of the sample presented in Figure 4.1.

it is expected that the model behaves with what was taught during the training, and with the erroneously labeled data, the model will have this behavior. The text model correctly classified the text, but the tweet contains irony and therefore it is possible to see that the FER model can help to overcome this problem. The only challenge is to find a dataset that is robust enough and properly labeled to be able to properly train the global image model.

4.5 Conclusion

It was possible to observe that from the tests, the effectiveness of using the 4 models together was proven, and discussed in section 4.4.
It was possible to observe a difference in the total number of images in the test set and in the amount presented in the confusion matrix, as there were images that generated errors in the execution of the model, so these cases were ignored. It was also observed that the YOLO model trained with the VOC dataset increases the test duration by about 3 hours, but its influence is relevant since in some cases the difference in accuracy was significant (for example, the results obtained in tests 4.3.3 and 4.3.5).
It was also possible to verify the effectiveness of the proposed model. However, if we discard the information from the salient areas, we can reach even higher values in accuracy, even though it is not possible to overcome the work in [LGAC21].
The B-T4SA dataset proved to be unreliable, as it was possible to observe situations in which the data labeling was done taking into account only the textual information.

Chapter 5

Conclusion

From this work, it was possible to understand the challenge of dealing with image sentiment classification. The literature research demonstrated to be an important part of this work because it allowed us to see what has been done and what we can do to innovate and improve the models in the area.

An idea for innovation was the use of a facial expression classifier in the model. Observing Table 2.12 we can see that this approach has not been employed yet, or wasn't used with three other models. After proposing a method to be developed, some concerns arose, mainly due to the time and resources available.

Each component has its role, and to work they have to operate together. From the tests, we can notice the proposed method obtained a good accuracy, 72.77%, overcoming the works done in [GA19b] and the state-of-the-art [VCC+17b]. However, ignoring the information from salient areas we obtained 80.86% accuracy. This can be justified since the architecture of the salient areas model is the same as the global image model, and due to the similar behaviour the results returned by them are unlikely to be different. Another factor is that the model was trained using the B-T4SA dataset, and we concluded that its labeling is not reliable. However, we can see advantages in using FER, for example, in a negative photo and an ironic caption, we can get around the irony by using visual information, and even if it is a clear image (that is, with features that indicate a positive image) if there is a sad face, the FER model will help to get around one more problem that may arise.

Due to the short time to carry out this project, since the literature research took a considerable amount of time, some options were taken in order to save time and be able to merge all the proposed models. Thus, it was considered to use, for the global image classifier, the architecture presented by [GA19b], since it overcame the state-of-the-art [VCC+17b] and therefore presented itself as an advantageous option.

For the salient areas classifier, the effort was focused on training the YOLO model, which would be responsible for detecting these areas. For classification, the global image classifier was maintained, since it made sense for the salient areas to be classified by the same classifier responsible for classifying the global images.

Since this project did not involve the development of a text classifier, not much time or effort was spent on this model. Initially, it was decided to use the Python libraries, as they are the fastest way to develop a small text classifier, in a way that would allow evaluating the behavior of the proposed method using the available textual information. However, the R-CNN text model, developed by [LGAC21], became the best option to be used.

The model that demanded the most time and effort was the FER model since its use with other models would be something new. A dataset was created, which contained images of

faces in controlled environments and in the wild, in order to present a variety during the model's training. This model obtained an accuracy of 72.75%.
During the realization of this project, we felt the need to create a dataset that was not labeled by machines, seen by the tests in Subsection 4.3.9 that the text classifier obtained a good result, and the example given in Section 4.4, which is understandable since the dataset used in the tests was tagged by AMT, which is not considered reliable [SW18, Mos]. However, it was possible to observe, when classifying the sample used as example, that the FER model can help to overcome the irony existing in the text. From the creation of the Twitter dataset, which is considerably small, containing 1,208 images, it was possible to get a sense of the time and effort needed to build a robust dataset, and which contained data from different social platforms (Facebook, Twitter, Reddit, Instagram, etc.), in order to enrich the training of the model for the prediction and identification of feelings in the analyzed posts.
But in this way, getting good accuracy with the tests, the key to taking the proposed method to another level would be the use of a reliable and robust dataset.

5.1 Contributions and Achievements

The main contribution is the method to process posts in social media, that includes a module for the facial expression recognition, and classifies their sentiment in order to enable the early detection of potentially violent events. Also, the creation of 2 datasets, one composed of images obtained directly from Twitter, which represent negative situations, and another composed of 50,783 face images. We also created a combined method using 4 base models that incorporated the FER classifier, and evaluated 3 approaches for fusing the information from the individual models.

5.2 Future Work

For future work, an important goal is the creation of a properly labeled dataset, in order to train the global image classifier. Another aspect to study is the improvement of the dataset used to train the FER model, since there is no specialization in the study of facial expressions, for example an expert in FACS, which is a comprehensive, anatomically based system for describing all visually discernible facial movement. It breaks down facial expressions into individual components of muscle movement, because some expressions were easily confused, such as fear and sadness.

The creation of a dataset with posts that precede an event, in order to improve the model's prediction for extreme events would also be an interesting follow up work, for example, collecting posts made before the protests of the yellow vest and trying to predict the following street demonstrations.

The creation of a larger dataset for the training of the YOLO model remains, and with the addition of more neutral classes (and not just negative ones), since some tests presented the YOLO model trained with the Twitter dataset as a useful model.

Instead of using 3 classes (negative, neutral, and positive), turning the model's output into a distribution of emotions remains as future work, and one of the sentiment models can be used, in order to detect extremist situations according to the distribution obtained.

Another possible future work would be to implement the observation of the classes of objects identified in the images and produce a distribution of emotions as a function of the objects detected, in order to influence the final classification.

Bibliography

[ACT+20] Alexandros Arjmand, Vasileios Christou, Alexandros Tzallas, Markos Tsipouras, Constantinos Angelis, Georgios Tsoumanis, E. Glavas, Roberta Forlano, Pinelopi Manousou, and Nikolaos Giannakeas. Transfer learning versus custom cnn architectures in nafld biopsy images. pages 480–483, 07 2020. 18

[ASM17] N Arunachalam, S Josephine Sneka, and G MadhuMathi. A survey on text classification techniques for sentiment polarity detection. In *2017 Innovations in Power and Advanced Computing Technologies (i-PACT)*, pages 1–5, 2017. 75

[Bas] S. Basaveswara. Cnn architectures, a deep-dive [online]. Available from: `https://towardsdatascience.com/cnn-architectures-a-deep-dive-a99441d18049` [cited 2020-11-05]. xv, 15, 16, 17

[BB12] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *J. Mach. Learn. Res.*, 13(null):281–305, February 2012. 52

[BDLM13] Igor Bisio, Alessandro Delfino, Fabio Lavagetto, and Mario Marchese. *Opportunistic Detection Methods for Emotion-Aware Smartphone Applications*, pages 53–85. 11 2013. xvi, 36

[BJC+13] Damian Borth, Rongrong Ji, Tao Chen, Thomas Breuel, and Shih-Fu Chang. Large-scale visual sentiment ontology and detectors using adjective noun pairs. In *Proceedings of the 21st ACM International Conference on Multimedia*, MM '13, page 223–232, New York, NY, USA, 2013. Association for Computing Machinery. Available from: `https://doi.org/10.1145/2502081.2502282`. 11

[BK17] R. Breuer and R. Kimmel. A deep learning perspective on the origin of facial expressions. *ArXiv*, abs/1705.01842, 2017. 32

[BL] Margaret M. Bradley and Peter J. Lang. Iaps message [online]. Available from: `https://csea.phhp.ufl.edu/media/iapsmessage.html` [cited 2020-12-03]. 3

[BL17] Margaret M. Bradley and Peter J. Lang. *International Affective Picture System*, pages 1–4. Springer International Publishing, Cham, 2017. Available from: `https://doi.org/10.1007/978-3-319-28099-8_42-1`. 3

[CBDC14] Tao Chen, Damian Borth, Trevor Darrell, and S. Chang. Deepsentibank: Visual sentiment concept classification with deep convolutional neural networks. *ArXiv*, abs/1410.8586, 2014. 41

[Cho20] Ambika Choudhury. Top 8 algorithms for object detection [online]. 2020. Available from: https://analyticsindiamag.com/top-8-algorithms-for-object-detection/ [cited 2020-12-14]. 19

[CL08] Manuel Calvo and Daniel Lundqvist. Facial expressions of emotion (kdef): Identification under different display-duration conditions. *Behavior research methods*, 40:109–15, 03 2008. 49

[CSGiNJ15] Victor Campos, Amaia Salvador, Xavier Giro-i Nieto, and Brendan Jou. Diving deep into sentiment. *Proceedings of the 1st International Workshop on Affect and Sentiment in Multimedia*, Oct 2015. Available from: http://dx.doi.org/10.1145/2813524.2813530. 55

[CWC+19] Yuedong Chen, Jianfeng Wang, Shikai Chen, Zhongchao Shi, and Jianfei Cai. Facial motion prior networks for facial expression recognition. *2019 IEEE Visual Communications and Image Processing (VCIP)*, pages 1–4, 2019. 69

[DBC] Tao Chen Damian Borth, Rongrong Ji and Shih-Fu Chang. Visual sentiment ontology [online]. Available from: http://visual-sentiment-ontology.appspot.com/ [cited 2020-12-03]. xv, 5, 6

[DeL] S. DeLand. When to use machine learning or deep learning? [online]. Available from: https://www.embedded-computing.com/guest-blogs/when-to-use-machine-learning-or-deep-learning [cited 2020-11-03]. 12

[des18] Fotos tiradas segundos antes do desastre [online]. 2018. Available from: https://www.paraoscuriosos.com/a7609/fotos-tiradas-segundos-antes-do-desastre [cited 2021-02-03]. xvi, 28

[dG] Universeté de Genève. Research material and online research [online]. Available from: https://www.unige.ch/cisa/research/materials-and-online-research/research-material/ [cited 2020-12-03]. 4

[DGLG11] Abhinav Dhall, Roland Goecke, Simon Lucey, and Tom Gedeon. Static facial expression analysis in tough conditions: Data, evaluation protocol and benchmark. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 2106–2112, 2011. 31, 32

[Dia19] G. Dias. Guilherme, um dos atiradores de massacre em escola de suzano, posta fotos com arma antes do crime [online]. 2019. Available from: https://cutt.ly/BkhWbdD [cited 2021-02-03]. xvi, 27

[DLHS16] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks, 2016. 20

[DM15] Shichuan Du and Aleix Martinez. Compound facial expressions of emotion: From basic research to clinical applications. *Dialogues in Clinical Neuroscience*, 17:443–455, 12 2015. 31

[DT05] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893 vol. 1, 2005. 19

[FCd19] Mathieu Fortin and Brahim Chaib-draa. Multimodal sentiment analysis: A multitask learning approach. In *ICPRAM*, pages 368–376, 01 2019. xvi, 11, 42, 45, 55

[FMP+19] Andrea Felicetti, Massimo Martini, Marina Paolanti, Roberto Pierdicca, Emanuele Frontoni, and Primo Zingaretti. *Visual and Textual Sentiment Analysis of Daily News Social Media Images by Deep Learning*, pages 477–487. 09 2019. 55

[GA19a] A. Gaspar and L. A. Alexandre. *Image Sentiment Analysis: Experimental Evaluation of Several Deep Learning Architectures*. 25th Portuguese Conference on Pattern Recognition (RECPAD 2019), Porto, October 31st, 2019. 11, 18, 39, 48, 55

[GA19b] António Gaspar and Luís Alexandre. *A Multimodal Approach to Image Sentiment Analysis*, pages 302–309. 10 2019. 11, 18, 38, 39, 55, 69, 81, 94, 95, 97

[GDDM14] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 580–587, 2014. 19

[GEC+13] Ian Goodfellow, Dumitru Erhan, Pierre Carrier, Aaron Courville, Mehdi Mirza, Ben Hamner, Will Cukierski, Yichuan Tang, David Thaler, Dong-Hyun Lee, Yingbo Zhou, Chetan Ramaiah, Fangxiang Feng, Ruifan Li, Xiaojie Wang, Dimitris Athanasakis, John Shawe-Taylor, Maxim Milakov, John Park, and Y. Bengio. Challenges in representation learning: A report on three machine learning contests. *Neural Networks*, 64, 07 2013. 31, 32

[GHP07] Gregory Griffin, Alex Holub, and Pietro Perona. Caltech-256 object category dataset. *CalTech Report*, 03 2007. 18

[Gir15] Ross B. Girshick. Fast r-cnn. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1440–1448, 2015. 19

[GMC+08] Ralph Gross, Iain Matthews, Jeffrey Cohn, Takeo Kanade, and Simon Baker. Multi-pie. In *2008 8th IEEE International Conference on Automatic Face Gesture Recognition*, pages 1–8, 2008. 31, 32

[Han] S. Handel. Classification of emotions. [online]. Available from: `https://www.theemotionmachine.com/classification-of-emotions/` [cited 2020-10-22]. 34, 35

[HCLW19] Yunxin Huang, Fei Chen, Shaohe Lv, and Xiaodong Wang. Facial expression recognition: A survey. *Symmetry*, 11:1189, 09 2019. xvi, xix, 27, 28, 29, 31, 32

[HG15] C.J. Hutto and Eric Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *ICWSM*, 01 2015. 75

[HKV19] Frank Hutter, Lars Kotthoff, and J. Vanschoren, editors. *Automatic machine learning: methods, systems, challenges*. Challenges in Machine Learning. Springer, Germany, 2019. 77

[HLVDMW17] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269, 2017. 52

[HLvdMW18] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks, 2018. xv, 16, 17

[HM17] B. Hasani and M. H. Mahoor. Facial expression recognition using enhanced deep 3d convolutional neural networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2278–2288, 2017. 32

[HRS+17] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, and Kevin Murphy. Speed/accuracy trade-offs for modern convolutional object detectors. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 3296–3297. IEEE Computer Society, 2017. Available from: `https://doi.org/10.1109/CVPR.2017.351`. xv, xvi, 22, 23, 24, 25, 26

[Hsa] Hsankesara. Flickr image dataset [online]. Available from: `https://www.kaggle.com/hsankesara/flickr-image-dataset` [cited 2020-12-09]. 10

[HTW+19] Steven C. Y. Hung, Cheng-Hao Tu, Cheng-En Wu, Chien-Hung Chen, Yi-Ming Chan, and Chu-Song Chen. Compacting, picking and growing for unforgetting continual learning. In *NeurIPS*, 2019. 69

[Hui18] Jonathan Hui. Object detection: speed and accuracy comparison (faster r-cnn, r-fcn, ssd, fpn, retinanet and yolov3) [online]. 2018. Available from: https://cutt.ly/NkhWxbI [cited 2020-12-16]. 21

[HZC21] Xin He, Kaiyong Zhao, and Xiaowen Chu. Automl: A survey of the state-of-the-art. *Knowledge-Based Systems*, 212:106622, 01 2021. 77

[HZRS14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *Lecture Notes in Computer Science*, page 346–361, 2014. Available from: http://dx.doi.org/10.1007/978-3-319-10578-9_23. 19

[HZRS16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 52

[HZZ+19] Feiran Huang, Xiaoming Zhang, Zhonghua Zhao, Jie Xu, and Zhoujun Li. Image-text sentiment analysis via deep multimodal attentive fusion. *Knowledge-Based Systems*, pages 26–37, 2019. 55

[KB14] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014. 53

[KCBW12] Kathrin Kaulard, Douglas W. Cunningham, Heinrich H. Bülthoff, and Christian Wallraven. The mpi facial expression database — a validated database of emotional and conversational facial expressions. *PLOS ONE*, 7(3):1–18, 03 2012. Available from: https://doi.org/10.1371/journal.pone.0032321. 31

[KK18] Evgeny Kim and Roman Klinger. A survey on sentiment and emotion analysis for computational literary studies. *ArXiv*, abs/1808.03137, 2018. xvi, 33, 35, 36

[KS16] Marie Katsurai and Shin'ichi Satoh. Image sentiment analysis using latent correlations among visual, textual, and sentiment views. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2837–2841, 2016. xix, 8, 11

[KSL19] Simon Kornblith, Jonathon Shlens, and Quoc V. Le. Do better imagenet models transfer better? *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2656–2666, 2019. 14

[LAE+16] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. *Lecture Notes in Computer Science*, page 21–37, 2016. Available from: http://dx.doi.org/10.1007/978-3-319-46448-0_2. 20

[LAKG98] Michael Lyons, Shigeru Akamatsu, Miyuki Kamachi, and Jiro Gyoba. Coding facial expressions with gabor wavelets. *Proceedings of the Third IEEE International Conference on Automatic Face and Gesture Recognition, April*, 1998:200 – 205, 05 1998. 31, 32

[LCK+10] P. Lucey, J. Cohn, T. Kanade, Jason M. Saragih, Z. Ambadar, and I. Matthews. The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*, pages 94–101, 2010. 31, 32

[LD19] Shan Li and Weihong Deng. Blended emotion in-the-wild: Multi-label facial expression recognition using crowdsourced annotations and deep locality feature learning. *International Journal of Computer Vision*, 127, 06 2019. 31

[LDD17] S. Li, W. Deng, and J. Du. Reliable crowdsourcing and deep locality-preserving learning for expression recognition in the wild. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2584–2593, 2017. 31

[LeD20] E. LeDell. H2o automl: Scalable automatic machine learning. 2020. 78

[LF□98] Daniel Lundqvist, Anders Flykt, and A. Öhman. The karolinska directed emotional faces – kdef, cd rom from department of clinical neuroscience, psychology section. *Karolinska Institutet*, pages 91–630, 01 1998. 31

[LGAC21] Vasco Lopes, António Gaspar, Luís A. Alexandre, and João Cordeiro. An automl-based approach to multimodal image sentiment analysis. *CoRR*, abs/2102.08092, 2021. Available from: `https://arxiv.org/abs/2102.08092`. xvi, 51, 52, 75, 81, 94, 95, 96, 97

[Li] C. Li. Transfer learning's best practice for image classification [online]. Available from: `https://cutt.ly/6khWhF8` [cited 2020-11-05]. 14

[LOB14] LOBI. Terms of use [online]. 2014. Available from: `https://exp.lobi.nencki.gov.pl/dnaps` [cited 2020-12-03]. 6

[Lor] S. Loria. Textblob: Simplified text processing [online]. Available from: `https://textblob.readthedocs.io/en/dev/` [cited 2021-04-22]. 75

[LW03] J Li and JZ Wang. Automatic linguistic indexing of pictures by a statistical modeling approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25:1075–1088, 09 2003. 18

[LZSC19] Y. Li, Jiabei Zeng, S. Shan, and X. Chen. Occlusion aware facial expression recognition using cnn with attention mechanism. *IEEE Transactions on Image Processing*, 28:2439–2450, 2019. 32

[Mar] P. Marcelino. Transfer learning from pre-trained models [online]. Available from: https://www.embedded-computing.com/guest-blogs/when-to-use-machine-learning-or-deep-learning [cited 2020-11-05]. xv, 13, 14

[Mat] Matter. Naps pictures [online]. Available from: http://www4.ujaen.es/~erpadial/NAPS.html [cited 2020-12-03]. 6

[MCM16] Ali Mollahosseini, David Chan, and Mohammad H. Mahoor. Going deeper in facial expression recognition using deep neural networks. *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Mar 2016. Available from: http://dx.doi.org/10.1109/WACV.2016.7477450. 32

[mer28] Merriam-webster online [online]. 1828. Available from: https://www.merriam-webster.com/ [cited 2020-10-21]. xix, 32, 33

[MGGTZC+20] Valeria Maeda-Gutiérrez, Carlos Galván Tejada, Laura Zanella Calzada, Jose Celaya Padilla, Jorge Galván Tejada, Hamurabi Gamboa-Rosales, Huizilopoztli Luna-Garcia, Rafael Magallanes-Quintanar, Guerrero-Mendez Carlos, and Carlos Olvera-Olvera. Comparison of convolutional neural network architectures for classification of tomato plant diseases. *Applied Sciences*, 10:1245, 02 2020. 18

[MH10a] Jana Machajdik and Allan Hanbury. Affective image classification using features inspired by psychology and art theory [online]. 2010. Available from: https://www.imageemotion.org/ [cited 2020-12-03]. xv, 4, 5

[MH10b] Jana Machajdik and Allan Hanbury. Affective image classification using features inspired by psychology and art theory. In *Proceedings of the 18th ACM International Conference on Multimedia*, MM '10, page 83–92, New York, NY, USA, 2010. Association for Computing Machinery. Available from: https://doi.org/10.1145/1873951.1873965. 3

[Mos] A. Moss. Concerns about bots on mechanical turk: Problems and solutions [online]. Available from: https://www.cloudresearch.com/resources/blog/concerns-about-bots-on-mechanical-turk-problems-and-solutions/ [cited 2021-06-02]. 98

[mov] Moves: Monitoring virtual crowds in smart cities [online]. Available from: http://moves.di.ubi.pt/ [cited 2020-10-21]. 2

[MSMSP14] Myriam Munezero, Calkin Suero Montero, Erkki Sutinen, and John Pajunen. Are they different? affect, feeling, emotion, sentiment, and opinion detection in text. *Affective Computing, IEEE Transactions on*, 5:101–111, 04 2014. xvi, 32, 33

[MZJG14] Artur Marchewka, Lukasz Zurawski, Katarzyna Jednoróg, and Anna Grabowska. The nencki affective picture system (naps): Introduction to a novel, standardized, wide-range, high-quality, realistic picture database. *Behavior research methods*, 06 2014. 6

[obs] Archaic" and "obsolete": What's the difference? [online]. Available from: `https://www.merriam-webster.com/words-at-play/whats-the-difference-between-archaic-and-obsolete` [cited 2020-10-21]. 32

[OFB20] Alessandro Ortis, Giovanni Farinella, and Sebastiano Battiato. Survey on visual sentiment analysis. *IET Image Processing*, 01 2020. xvi, 3, 4, 6, 46, 48

[OFTB18] Alessandro Ortis, Giovanni M. Farinella, Giovanni Torrisi, and Sebastiano Battiato. Visual sentiment analysis based on on objective text description of images. In *2018 International Conference on Content-Based Multimedia Indexing (CBMI)*, pages 1–6, 2018. 55

[PCSG15] Kuan-Chuan Peng, Tsuhan Chen, Amir Sadovnik, and Andrew Gallagher. A mixed bag of emotions: Model, predict, and transfer emotion distributions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 860–868, 2015. xv, 7, 8, 9, 11

[Pim] T. Pimenta. Conheça todos os tipos de neurotransmissores e saiba porque eles são importantes para sua saúde. [online]. Available from: `https://www.vittude.com/blog/neurotransmissores/` [cited 2020-10-22]. xvi, 36, 37

[PSGC16] Kuan-Chuan Peng, Amir Sadovnik, Andrew Gallagher, and Tsuhan Chen. Where do emotions come from? predicting the emotion stimuli map. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 614–618, 2016. xv, 9

[PVRM05] M. Pantic, M. Valstar, R. Rademaker, and L. Maat. Web-based database for facial expression analysis. In *2005 IEEE International Conference on Multimedia and Expo*, pages 5 pp.–, 2005. 31, 32

[PY10] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010. 53

[Qui19] S. Quintella. Atirador de suzano teria postado trinta fotos antes de invadir escola [online]. 2019. Available from: `https://vejasp.abril.com.br/cidades/atirador-suzano-facebook/` [cited 2021-02-03]. 27

[RB99] James Russell and Lisa Barrett. Core affect, prototypical emotional episodes, and other things called emotion: Dissecting the elephant. *Journal of personality and social psychology*, 76:805–19, 06 1999. 35

[RDGF16] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016. 20

[RHGS15] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 91–99, Cambridge, MA, USA, 2015. MIT Press. 20

[Rus03] James Russell. Core affect and the psychological construction of emotion. *Psychological review*, 110:145–72, 02 2003. 35

[RW17] Waseem Rawat and Zenghui Wang. Deep convolutional neural networks for image classification: A comprehensive review. *Neural Computation*, 29:1–98, 06 2017. 13

[Sad] Amir Sadovnik. Advanced multimedia processing (amp) lab [online]. Available from: `http://chenlab.ece.cornell.edu/downloads.html` [cited 2020-12-03]. 7, 9

[Sha] P. Sharma. 7 popular image classification models in imagenet challenge (ilsvrc) competition history [online]. Available from: `https://cutt.ly/tkhQ6cm` [cited 2020-11-06]. 18

[sim19] Simpson dataset [online]. 2019. Available from: `https://vrai.dii.univpm.it/content/simpson-dataset` [cited 2021-02-03]. 55

[Siy] B. Siyah. Imagenet winning cnn architectures (ilsvrc). [online]. Available from: `https://www.kaggle.com/getting-started/149448` [cited 2020-11-06]. 18

[SMK17] Courtney Spoerer, Patrick McClure, and Nikolaus Kriegeskorte. Recurrent convolutional neural networks: A better model of biological object recognition. *Frontiers in Psychology*, 8:1551, 09 2017. 38

[SMW20] Henrique Siqueira, S. Magg, and S. Wermter. Efficient facial feature learning with wide ensemble-based convolutional neural networks. In *AAAI*, 2020. 69

[SP18] Manali Shaha and M. Pawar. Transfer learning for image classification. *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, pages 656–660, 2018. 18

[STD14] P. Suja, Shikha Tripathi, and J. Deepthy. Emotion recognition from facial expressions using frequency domain techniques. In Sabu M. Thampi, Alexander Gelbukh, and Jayanta Mukhopadhyay, editors, *Advances in*

Signal Processing and Intelligent Recognition Systems, pages 299–310, Cham, 2014. Springer International Publishing. xvi, 49, 50

[SW18] C. Stokel-Walker. Bots on amazon's mechanical turk are ruining psychology studies [online]. 2018. Available from: https://www.newscientist.com/article/2176436-bots-on-amazons-mechanical-turk-are-ruining-psychology-studies/#ixzz6x4HKCdRS [cited 2021-06-03]. 98

[SYLM18] Kaikai Song, Ting Yao, Qiang Ling, and Tao Mei. Boosting image sentiment analysis with visual attention. *Neurocomputing*, 312:218 – 228, 2018. Available from: http://www.sciencedirect.com/science/article/pii/S092523121830701X. 55

[SYWS16] Ming Sun, Jufeng Yang, Kai Wang, and Hui Shen. Discovering affective regions in deep convolutional neural networks for visual sentiment prediction. In *2016 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, 2016. 55

[TC18] Hung-Hsu Tsai and Yi-Cheng Chang. Facial expression recognition using a combination of multiple facial features and support vector machine. *Soft Computing*, 22:1–17, 07 2018. 32

[Ult] Ultralytics. Yolov5 [online]. Available from: https://github.com/ultralytics/yolov5 [cited 2021-04-12]. xvii, 70

[Vad] Lucia Vadicamo. Cross-media learning for image sentiment analysis in the wild [online]. Available from: http://www.t4sa.it/ [cited 2020-12-08]. 10

[Var20] A. Vargas. Um ano após ataque em escola em suzano, túmulo de assassino recebe visitas de admiradores [online]. 2020. Available from: https://www.bbc.com/portuguese/brasil-51880555 [cited 2021-02-03]. 27

[VCC+17a] L. Vadicamo, F. Carrara, A. Cimino, S. Cresci, F. Dell'Orletta, F. Falchi, and M. Tesconi. Cross-media learning for image sentiment analysis in the wild. In *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 308–317, 2017. 9, 11, 39

[VCC+17b] Lucia Vadicamo, Fabio Carrara, Andrea Cimino, Stefano Cresci, Felice Dell'Orletta, Fabrizio Falchi, and Maurizio Tesconi. Cross-media learning for image sentiment analysis in the wild. In *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 308–317, 2017. 11, 55, 69, 94, 95, 97

[VDDP18] Athanasios Voulodimos, Nikolaos Doulamis, Anastasios Doulamis, and Eftychios Protopapadakis. Deep learning for computer vision: A brief

review. *Computational Intelligence and Neuroscience*, 2018:1–13, 02 2018. xv, 13, 15

[VLYK20] T. Vo, G. Lee, H. Yang, and S. Kim. Pyramid with super resolution for in-the-wild facial expression recognition. *IEEE Access*, 8:131988–132001, 2020. 69

[WLFM09] Jacob Whitehill, Gwen Littlewort, Ian Fasel, and Javier Movellan. Toward practical smile detection. *IEEE transactions on pattern analysis and machine intelligence*, 31:2106–11, 11 2009. 31

[WMW20] Zhuanghui Wu, M. Meng, and J. Wu. Visual sentiment prediction with attribute augmentation and multi-attention mechanism. *Neural Processing Letters*, 51:2403–2416, 2020. 5

[WPY+19] Kai Wang, Xiaojiang Peng, Jianfei Yang, Debin Meng, and Yu Qiao. Region attention networks for pose and occlusion robust facial expression recognition. *CoRR*, abs/1905.04075, 2019. Available from: `http://arxiv.org/abs/1905.04075`. 69

[WQJZ20a] Lifang Wu, Mingchao Qi, Meng Jian, and Heng Zhang. Visual sentiment analysis by combining global and local information. *Neural Processing Letters*, 51:1–13, 06 2020. 11

[WQJZ20b] Lifang Wu, Mingchao Qi, Meng Jian, and Heng Zhang. Visual sentiment analysis by combining global and local information. *Neural Processing Letters*, 51:1–13, 06 2020. 41, 55

[WRP+17] Robert Walecki, Ognjen Rudovic, Vladimir Pavlovic, Björn W. Schuller, and Maja Pantic. Deep structured learning for facial action unit intensity estimation. *CoRR*, abs/1704.04481, 2017. Available from: `http://arxiv.org/abs/1704.04481`. 30

[YCY18] H. Yang, U. Ciftci, and L. Yin. Facial expression recognition by de-expression residue learning. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2168–2177, 2018. 32

[YLJY15a] Quanzeng You, Jiebo Luo, Hailin Jin, and Jianchao Yang. Robust image sentiment analysis using progressively trained and domain transferred deep networks. 09 2015. 11

[YLJY15b] Quanzeng You, Jiebo Luo, Hailin Jin, and Jianchao Yang. Robust image sentiment analysis using progressively trained and domain transferred deep networks. In Blai Bonet and Sven Koenig, editors, *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA*, pages 381–388. AAAI Press, 2015. Available from: `http://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9556`. 11, 37, 41, 55

[YSS17] Jufeng Yang, Ming Sun, and Xiaoxiao Sun. Learning visual sentiment distributions via augmented conditional probability neural network, 2017. Available from: https://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14506. 55

[YSS+18] Jufeng Yang, Dongyu She, Ming Sun, Ming-Ming Cheng, Paul L. Rosin, and Liang Wang. Visual sentiment prediction based on automatic discovery of affective regions. *IEEE Transactions on Multimedia*, 20:2513–2525, 2018. 41

[YWS+06] Lijun Yin, Xiaozhou Wei, Yi Sun, Jun Wang, and M.J. Rosato. A 3d facial expression database for facial behavior research. In *7th International Conference on Automatic Face and Gesture Recognition (FGR06)*, pages 211–216, 2006. 31, 32

[YZY18] H. Yang, Z. Zhang, and L. Yin. Identity-adaptive facial expression recognition through expression regeneration using conditional generative adversarial networks. In *2018 13th IEEE International Conference on Automatic Face Gesture Recognition (FG 2018)*, pages 294–301, 2018. 32

[ZDH+18] Sicheng Zhao, Guiguang Ding, Qingming Huang, Tat-Seng Chua, Björn W. Schuller, and Kurt Keutzer. Affective image content analysis: A comprehensive survey. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 5534–5541. International Joint Conferences on Artificial Intelligence Organization, 7 2018. Available from: https://doi.org/10.24963/ijcai.2018/780. 7

[Zha] Sicheng Zhao. Sicheng zhao [online]. Available from: https://sites.google.com/site/schzhao/ [cited 2020-12-03]. 7

[ZHT+11] Guoying Zhao, Xiaohua Huang, Matti Taini, Stan Li, and Matti Pietikäinen. Facial expression recognition from near-infrared videos. *Image Vision Comput.*, 29:607–619, 08 2011. 31, 32

[ZWS+20] Hongbin Zhang, Jinpeng Wu, Haowei Shi, Ziliang Jiang, Donghong Ji, Yuan Tian, and Guangli Li. Multidimensional extra evidence mining for image sentiment analysis. *IEEE Access*, PP:1–1, 06 2020. 11, 12, 40, 48, 55

[ZYG+16a] Sicheng Zhao, Hongxun Yao, Yue Gao, Guiguang Ding, and Tat-Seng Chua. Predicting personalized image emotion perceptions in social networks. *IEEE Transactions on Affective Computing*, PP:1–1, 11 2016. 7

[ZYG+16b] Sicheng Zhao, Hongxun Yao, Yue Gao, RongRong Ji, and Guiguang Ding. Continuous probability distribution prediction of image emotions via

multi-task shared sparse regression. *IEEE Transactions on Multimedia*, PP:1–1, 10 2016. 7, 8

[ZZLQ16] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, 2016. xvii, 58, 59, 60