UNIVERSITY OF THE PELOPONNESE & NCSR "DEMOCRITOS"
MSC PROGRAMME IN DATA SCIENCE

Movie shot classification using machine learning

by

Apostolos Maniatis

A thesis submitted in partial fulfillment
of the requirements for the MSc
in Data Science

Supervisor: Theodoros Giannakopoulos
Principal Researcher of Multimodal Machine Learning

Athens, June 2021

Apostolos Maniatis

MSc. Thesis, MSc. Programme in Data Science

University of the Peloponnese & NCSR "Democritos", June 2021

UNIVERSITY OF THE PELOPONNESE & NCSR "DEMOCRITOS"
MSC PROGRAMME IN DATA SCIENCE

Movie shot classification using machine learning

by

Apostolos Maniatis

A thesis submitted in partial fulfillment
of the requirements for the MSc
in Data Science

Supervisor: Theodoros Giannakopoulos
Principal Researcher of Multimodal Machine Learning

Approved by the examination committee on June, 2021.

(Signature)	(Signature)	(Signature)
.....	.....	.....
Theodoros Giannakopoulos Principal Researcher	Georgios Petasis Associate Researcher	Ioannis Moscholios Assistant Professor

Athens, June 2021

UNIVERSITY OF THE PELOPONNESE & NCSR "DEMOCRITOS"
MSC PROGRAMME IN DATA SCIENCE

Declaration of Authorship

(1) I declare that this thesis has been composed solely by myself and that it has not been submitted, in whole or in part, in any previous application for a degree. Except where states otherwise by reference or acknowledgment, the work presented is entirely my own.

(2) I confirm that this thesis presented for the degree of Bachelor of Science in Informatics and Telecommunications, has

(i) been composed entirely by myself

(ii) been solely the result of my own work

(iii) not been submitted for any other degree or professional qualification

(3) I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or processional qualification except as specified.

(Signature)

.....

Apostolos Maniatis

Athens, June 2021

Acknowledgments

Throughout the writing of this thesis, I have received a great deal of support and assistance. I would first like to thank my supervisor, Professor Theodoros Giannakopoulos, whose expertise was invaluable in formulating the research questions and methodology. Without his guidance and persistent help, this dissertation would not have been possible.

I would also like to thank a classmate Electra Sifacaki with which I worked on a part of this thesis. There was mutual understanding, and our cooperation was profitable and effective. Moreover I would like to thank all anonymous annotators for their participation in the video annotation process to create the final dataset. Finally, I would like to express my gratitude to my parents and my brothers. Without their tremendous understanding and encouragement in the past few years, it would be impossible to complete my study.

Περίληψη

Στην παραγωγή ταινιών shot, ονομάζουμε την χρονική ενότητα κατά την οποία κυριαρχεί ένα συγκεκριμένο και συνεχές είδος κίνησης της κάμερας μέχρι να ξεκινήσει μία επόμενη ενότητα με διαφορετικό τύπο κίνησης ή απο διαφορετική κάμερα. Το shot δεν σχετίζεται με την σκηνή η οποία έχει ευρύτερα χρονικά, χωρικά και σημασιολογικά χαρακτηριστικά. Μία ταινία αποτελείται από έναν μεγάλο αριθμό από shots όπου μπορεί να διαφέρουν μεταξύ τους σκηνοθετικά. Η παρούσα διπλωματική εργασία έχει ως στόχο την ανίχνευση των shots που παρουσιάζονται σε μία ταινία και την ταξινόμηση του κάθε ενός από αυτά στην κλάση στην οποία ανήκει. Η διαδικασία της ταξινόμησης γίνεται μέσω αλγορίθμων επιβλεπόμενης μηχανικής μάθησης. Επιπλέον, παρουσιάζεται ένα νέο σύνολο δεδομένων το οποίο περιέχει shots ταξινομημένα σε διάφορες σκηνοθετικές κατηγορίες, και μία αντίστοιχη διαδικασία επισημείωσης σύμφωνα με την οποία έχουν απαιτηθεί δεδομένα από τρείς τουλάχιστον χρήστες. Τέλος στην παρούσα εργασία παρουσιάζεται ένα demo στο οποίο γίνεται η αξιολόγηση των εκπαιδευομένων αλγορίθμων σε πραγματικές ταινίες μέσω στατιστικών συσχετίσεων των αποτελεσμάτων κατηγοριοποίησης των επιμέρους shots.

Abstract

In movie production, we call shot the time unity upon which a specific and continuous type of camera is dominated until a subsequent module with a different type of movement or from another camera begins. A shot is not related to the scene that has wider time, spatial and semantic characteristics. A movie consists of a large number of shots where they can differ from each other-directed. This work aims to detect shots presented in a film and classify each of them in the class to which it belongs. The classification process is through supervised machine learning algorithms. In addition, a new dataset containing shots is presented classified in various directorial categories and a corresponding labeling process that data from at least three users have been required. Finally, this work presents a demo to evaluate trained algorithms in real movies through statistical associations of the categorization results of the individual shots.

Contents

List of Tables iii

List of Figures iv

List of Abbreviations vi

1 Introduction 1

1.1 Problem description 1

1.2 Thesis structure 3

2 Background Methodology 5

2.1 Supervised Learning 5

2.1.1 Classification metrics 5

2.1.2 Classification algorithms 7

2.2 Video feature extraction 10

3 Dataset Description 15

3.1 Overview 15

3.2 Shot generation 16

3.3 Shot classes 19

3.4 Shot annotation 21

3.5 Annotation data aggregation 23

4 Experiments 27

4.1 Shot classification performance 27

4.2 Qualitative evaluation and demo 39

5 Conclusions and Future Work 45

5.1 Conclusions 45

5.2 Feature work 45

List of Tables

2.1 Confusion matrix for binary classification 6

3.1 Values of parameters for hyperparameter tuning 17

3.2 Results of the second phase of hyperparameter tuning 18

3.3 Number of samples per class of the final dataset 26

4.1 Number of samples per class 28

4.2 Performance of classifiers in Binary classification task 28

4.3 Performance of classifiers in 3 classes classification task 32

4.4 Performance of classifiers in 4 classes classification task 34

4.5 Performance of classifiers in 9 classes classification task 36

4.6 Performance of ET classifier with Delta features 37

4.7 Predictions of trained models to movies in Binary classification task 40

4.8 Predictions of trained models to movies in 3 class classification task 41

4.9 Predictions of trained models to movies in 4 class classification task 41

4.10 Predictions of trained models to movies in 9 class classification task 42

4.11 Clustering predictions 43

List of Figures

2.1 Aplha and Total Error[20] 9

2.2 Conceptual diagram of the visual feature extraction process 12

3.1 Recall, Precision and F1 scores of experiments in hyperparameter tuning process 18

3.2 Annotation page of web application 22

3.3 User annotation page of web application 22

3.4 (a) Number of samples before users' agreement.(b) Number of samples after users' agreement 24

3.5 Confusion matrix of users' agreement 25

4.1 (a) Histogram of a red feature between static and non-static classes.(b) Histogram of a green feature between static and non-static classes.(c) Histogram of a blue feature between static and non-static classes 30

4.2 Confusion matrix and ROC curves of every class in the binary classification task. 31

4.3 Confusion matrix and ROC curves of every class in the 3 classes classification task 33

4.4 Confusion matrix and ROC curves of every class in the 4 classes classification task 35

4.5 Confusion matrix and ROC curves of every class in the 9 classes classification task 38

4.6 Plots of clusters for each classification task 44

List of Abbreviations

URI	Uniform Resource Identifier
YAML	Yet Another Markup Language
i.e.	id est
e.g	exempli gratia
LSTM	Long Short Term Memory
KNN	K-Nearest Neighbor
SVM	Support-vector machine
AUC	Area Under Curve
ROC	Receiver Operating Characteristic
RGB	Red Green Blue
RF	Random Forest
DT	Decision Trees
ET	Extra Trees
AB	Adaboost
SMOTE	Synthetic Minority Oversampling Technique

Chapter 1

Introduction

1.1 Problem description

In recent years the classification of different types of videos in plenty of classes is an active field of research in the Machine and Deep Learning industries. The main reason is the possible implementation in several areas, such as video retrieval, personalized video search, video summarization, etc. The evolution of the technology has allowed the recording and creation of various types of videos very effortlessly. Anyone now using a mobile phone can easily record what happens in his everyday life. In this way, video creation has become a daily habit.

In addition, a field that revolutionized entertainment and first appeared in 1878 is the film industry. In recent years, an exponential increase in films is observed, specifically in the year 2.000, the number of movies released in the United States and Canada was about 371, while in 2019, they reached up to 792. [1] [2] Therefore we can confidently say that it is now a field of demand, and film production is constantly increasing.

Most movies are available online. There are many online movie streaming services such as Netflix (`https://www.netflix.com`), Disney Plus(`https://www.disneyplus.com`), HBO Max (`https://www.hbomax.com`), etc., where each user can choose to see a wide range of different movies. Most users now seem to prefer to be members of these online streaming services because they choose to see whatever

movie they want the moment they want. For example, in April 2021, Netflix published 207.64 million paid subscribers worldwide,[3] while Disney+ has 103 Million Global Subscribers as of April 2021. [4]

There is a large volume of movies available for users on all these platforms. When connected to the graphical environment of such a platform, the user is lost in plenty of films. This affects that the user has difficulty take a choice of which movie to watch. In this way, the film selection consists of a time-consuming process and ends up being tedious. For this purpose, there is a need for a recommendation system. In particular, a system that will propose movies to the user based on the films he has seen previously.

The films consist of various shots (videos), merged and synthesizing a final long-term video. In this work, we propose a methodology that classifies these shots into different categories depending on the camera movement. Through camera movement, we can take essential conclusions about a movie. The movement of a camera creates different viewer feelings like fear, anxiety, emotion, etc. In this way, we can assume the aesthetic and the type of a movie. For example, if there are quick camera movements, this has resulted in anxiety and nervousness to be created in the viewer, so we can assume that it is an action movie. Thus, we can take conclusions about the direction of a film by classifying the shots and therefore presume who the director may be.

A critical problem that someone can meet in the classification of a shot to a class, is to know what classes are available. So a survey should be done to find all available types of shots that exist. It is then essential to find an accurate way of identifying these shots, namely, from a whole movie, to recognize when shots are changed. This is a challenging task because we need to find what parameters (e.g., brightness, the difference in black and white) play a significant role in determining the shots change. To succeed in classifying a video, we need to know what features we have to extract, that is, which features from a video will be useful for recognition to be more accurate. So it should follow a feature extraction process and then training various algorithms to find out what features are useful for classifying a video to a class and which features may be unnecessary.

In this work, we propose using supervised knowledge from visual domains to achieve the classification of shots in various categories. In particular, it follows the recognition of shots from a given video and then the separation of these to individual videos. Videos are sorted between 15 categories, and we use six different supervised classifiers to classify them. Moreover, we present a novel dataset comprising from real movies. We also present in detail how we created our dataset, explaining the entire annotation process carried out by a group of human annotators, using an annotating tool created for this work's purposes.

1.2 Thesis structure

The rest of this thesis is organized as follows: In Chapter 2, we present the entire theory background based on this thesis. Explains basic concepts such as Supervised Learning, which algorithms we used to train our models, and a summary description about them. We also analyze the whole process of feature extraction, explaining what features we extracted. Then in Chapter 3, we present and illustrate the classes of shots along with examples to better understand them. Describe the whole process that followed the shot detection and then the shot generation. In detail, we report all the steps we followed to create our own dataset from scratch. Chapter 4 presents all the experiments we made and the results along with a quality assessment in real movies, to have a better picture of our model's performance. Conclusions and feature work are drawn in section 5.

Chapter 2

Background Methodology

2.1 Supervised Learning

Supervised Learning is a field of Machine Learning. The task of Supervised Learning is given a training set of N examples $(x_1, y_1), ...(x_N, y_N)$, where each yj came from an unknown function $y = f(x)$ to find a function h that approaches better the true function f. The function h is a hypothesis.[5] Essentially, the learner who will find the appropriate h function accepts some labeled examples and, after following the training process, making predictions to unseen data.[6]

Supervised Learning consists of two primary sectors Classification and Regression. The difference between them is that Classification is a task of approximating a mapping function (f) from input variables (x) to discrete output variables (y). In contrast, in the Regression task, the output variable (y) is continuous.[7]

2.1.1 Classification metrics

The quality of a learning algorithm on whether it has learned the training data results from recognizing data that has not been seen again. The statistical quality of an algorithm for whether it has learned to identify new data other than training data is called generalization error. [8] In classification, there are some metrics widely used to measure the performance of various training algorithms. A widespread metric is the accuracy that is usually the starting point. Accuracy is the number of correct

Table 2.1: Confusion matrix for binary classification

		Predicted	
		Positive	Negative
Actual	Positive	TP	FN
	Negative	FP	TN

predictions made divided by the total number of predictions. However, because accuracy does not always give us a comprehensive picture of the performance of our models, other metrics give us further information. Precision is the fraction of relevant instances among the retrieved examples, while Recall is the fraction of relevant instances retrieved.[9] Finally, there is the f1 score where is the weighted average of precision and recall.

Still, a very popular measure used for binary and multiclass classification is the Confusion matrix. The confusion matrix represents counts from predicted and actual values.[10] An example of a confusion matrix for binary classification is shown in Table 2.1. The output True Positive 'TP' indicates the number of examples correctly classified or detected. The term False Positive 'FP' is incorrectly classified or detected, e.g., the number of negative examples classified as positive. False negative 'FN' is the wrongly rejected examples, and the True negative 'TN' the exampled that rejected correctly.[11] Having all these above items, we have a very good idea of the performance of our model. We can understand which class learned the learning algorithm at a satisfactory level and which class has difficulty recognizing.

One way to have a visual image of our model performance is the receiver operating characteristic (ROC) curve. A ROC curve is a graph showing the performance of a classification model [12] for several different classification threshold values between 0.0 and 1.0. This curve plots two parameters, sensitivity or True Positive Rate (TPR) and False Positive Rate (FPR). Sensitivity is the number of instances from the positive class that are predicted correctly. More specifically, it describes how good the model predicts the positive class when the actual outcome is positive. On the other hand, FPR summarizes how often a positive class is predicted when the actual outcome is negative. The ROC graph summarises the confusion matrices produced for each threshold without having actually to calculate them. Just by

glancing over the graph, we can conclude that one threshold is better than another and depending on how many false positives we are willing to accept, we can choose the optimal threshold.

2.1.2 Classification algorithms

In the history of machine learning, a great many classification algorithms have been proposed. The following algorithms are more widely used and have proven to be stable and bring satisfactory results.

1. **Decision trees** (DT) [13] incorporate a supervised classification approach. DT are a hierarchical data structure that represents data using a divide-and-conquer strategy. A tree is composed of nodes, branches, and endpoints. Each node represents a point at which a decision has to be taken. The branches emanating from nodes are the alternatives from which a choice can be selected. Each endpoint of the tree has an associated value which is the pay-off from reaching that endpoint.

2. **Support Vector Machines** (SVMs) [14] is a supervised algorithm for classification and regression tasks. Concerning classification tasks, SVM builds a hyperplane that separates instances of different classes within a varying margin and the nearest data samples, which called support vectors. The margin is calculated, when two parallel hyperplanes are positioned on each side of the separated hyperplance. The parallel hyperplanes are calculated to distinguish the two classes. A good generalisation for SVM is achieved when the distance of support vectors for each class is maximised from the margin, as the subjected separation reduces the risk of a sample to be categorised erroneously.

3. **K-Nearest Neighbor** (KNN) [15] is a very intuitive method that classifies unlabeled examples based on their similarity with examples in the training set. KNN captures the idea of similarity by calculating the distance between points on a graph. There are several ways to calculate the distance between the data points. A commonly used distance metrics for continuous variables is Euclidean distance and Manhattan distance.

4. **Random Forest** (RF) Ensemble learning classifiers have draw increasing attention because they are more accurate. Many authors have demonstrated significant performance improvement, Random Forests (RF) introduced by Breiman [16], an ensemble learning method for classification, which operates by combining several univariate decision trees to build an ensemble that uses the whole forest compositing the classifier model. The collection of tree-structured classifiers $\{h(x, \Theta_k), k = 1, \dots\}$ where $\{\Theta_k\}$ are independent identically distributed random vectors and each tree casts a vote for the assignation of the most frequent class to the input at input x. Hence, some data may be used more than once in the training of classifiers, while others might never be used. The classification label a new sample that is decided by aggregating the predictions of the trees in the ensemble through majority voting. A RF increases the diversity of the trees by making them grow from different training data subsets created with bagging or bootstrap aggregation.[17] Bootstrap aggregation or bagging is a method used for training data with random resampling with replacement, on the original data set. Each subset selected using bagging to make each individual bth tree grow usually contains 2/3 of the test dataset. The samples which are not present in the test subset are included as part of another subset called out-of-bag (oob). A different oob subset is created for every bth tree from the unselected samples during bagging process. Concerning feature selection, tree design requires to choose the proper measure to maximise dissimilarity between classes. There are many techniques for building a tree and selecting the attributes such gain ratio and Gini index. In our tested framework RF uses by default Gini index as a measure for best split selection, which measures the impurity of a given attribute respect to the rest of the classes.

5. **Extremely randomized trees** or Extratrees [18] is a tree-based ensemble method for supervised classification and regression problems. Extratrees works by creating a large number of unpruned decision trees from the training dataset. Predictions are made by averaging the prediction of the decision trees in the case of regression or using majority voting in classification. Unlike

bagging and random forest that develop each decision tree from a bootstrap sample of the training dataset, the Extra Trees algorithm fits each decision tree on the whole training dataset. They are the number of decision trees in the ensemble, the number of input features to select and consider for each split point randomly, and the minimum number of samples required in a node to create a new split point.

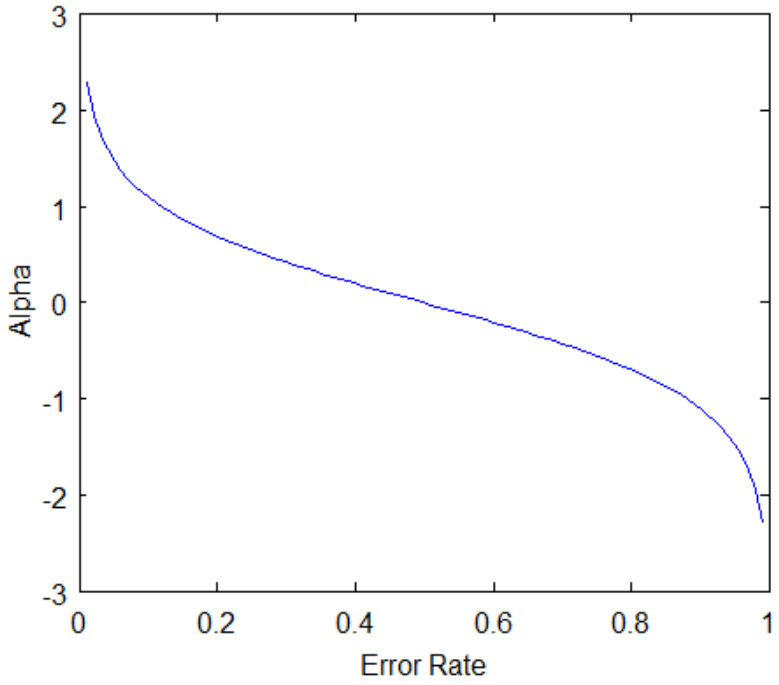


Figure 2.1: Aplha and Total Error[20]

6. **Adaboost** belongs to boosting Ensembles algorithms and first occurred by Yoav Freund and Robert Schapire.[19] The general idea of boosting is to improve the predictive flexibility of simple models. In particular, it trains a large number of "Weak" learners (classifiers) in sequence. Initially, training a simple model from training data; afterwards, training a second model aims to improve the errors of the previous model. Continue models are created until the general algorithm predicts the training set to a satisfactory level or reaches the upper limit of the models. The final model results from two key factors alpha and total error. Alpha expresses how much influence a stump will have

in the final classification and is calculated by the formula:

$$a_t = \frac{1}{2}{}^{ln}\frac{(1 - TotalError)}{TotalEroor} \tag{2.1}$$

Total Error is the total number of misclassifications for that training set divided by the training set size. As we see in Figure 2.1 when the Alpha is positive, the predicted and the actual output agree. On the contrary, if the Alpha is negative, The predicted output does not agree with the actual class i.e. the sample is misclassified. [21]

2.2 Video feature extraction

Feature extraction is a core component of the computer vision pipeline. In computer vision, a feature is a measurable piece of data that is unique to a specific object. It may be a distinct color in an image or a particular shape, such as a line, edge, e.g. So a video can be represented with a number of hand-crafted features that, if selected correctly, can represent the contents of a whole video in detail.

We extracted hand-crafted features applied to several tasks such as visual classification and clustering to achieve feature representation. Our primary goal was to extract as many visual elements as possible. We have adopted a wide range of visual features to describe the content of the visual information. All the features that we extracted represent the visual characteristics of the video.

In particular, every 0.2 sec, the following 88 visual features are extracted from the corresponding frame:

- Color - related features (45 features):
  - 8-bin histogram of the red values
  - 8-bin histogram of the green values
  - 8-bin histogram of the blue values
  - 8-bin histogram of the grayscale values

  - 5-bin histogram of the max-by-mean-ratio for each RGB triplet
  - 8-bin histogram of the saturation values

- Average absolute difference between two successive frames in grey scale (1 feature)

- Facial features (2 features): The Viola-Jones [22] OpenCV implementation is used to detect frontal faces and the following features are extracted per frame:
  - number of faces detected
  - average ratio of the faces' bounding boxes areas divided by the total area of the frame

- Optical-flow related features (3 features): The optical flow is estimated using the 285 Lucas-Kanade method [24] and the following 3 features are extracted:
  - average magnitude of the flow vectors
  - standard deviation of the angles of the flow vectors
  - a hand-crafted feature that measures the possibility that there is a camera tilt movement – this is achieved by measuring a ratio of the magnitude of the flow vectors by the deviation of the angles of the flow vectors.

- Current shot duration (1 feature): a basic shot detection method is implemented in this library. The length of the shot (in seconds) in which each frame belongs to, is used as a feature.

- Object-related features (36): We use the Single Shot Multibox Detector [23] method for detecting 12 categories of objects. For each frame, as soon as the object(s) of each category are detected, three statistics are extracted: number of objects detected, average detection confidence and average ratio of the objects' area to the area of the frame. So in total, 3x12=36 object-related features are extracted. The 12 object categories we detect are the following: person, vehicle, outdoor, animal, accessory, sports, kitchen, food, furniture, electronic, appliance and indoor

In addition, we computed six video-level statistics of the above non-object features. In particular, mean, std, median by std ratio, top-10 percentile. As for the object detection, the frame-level predictions are post-processed under local time windows in two different ways: (i) the object frame-level confidences are smoothed across time windows in order to increase the accuracy of the predictions and (ii) every object that is not present to at least a minimum number (threshold) of subsequent frames, is excluded from the final feature vector. However, this smoothing procedure is the only post-processing performed on the object-related features: no other statistics are extracted for the whole video, other than the object features' simple averages. This process, therefore, results in 243 feature statistics that describe the entire video. Figure 2.2 shows the conceptual diagram of the process followed to extract visual features.

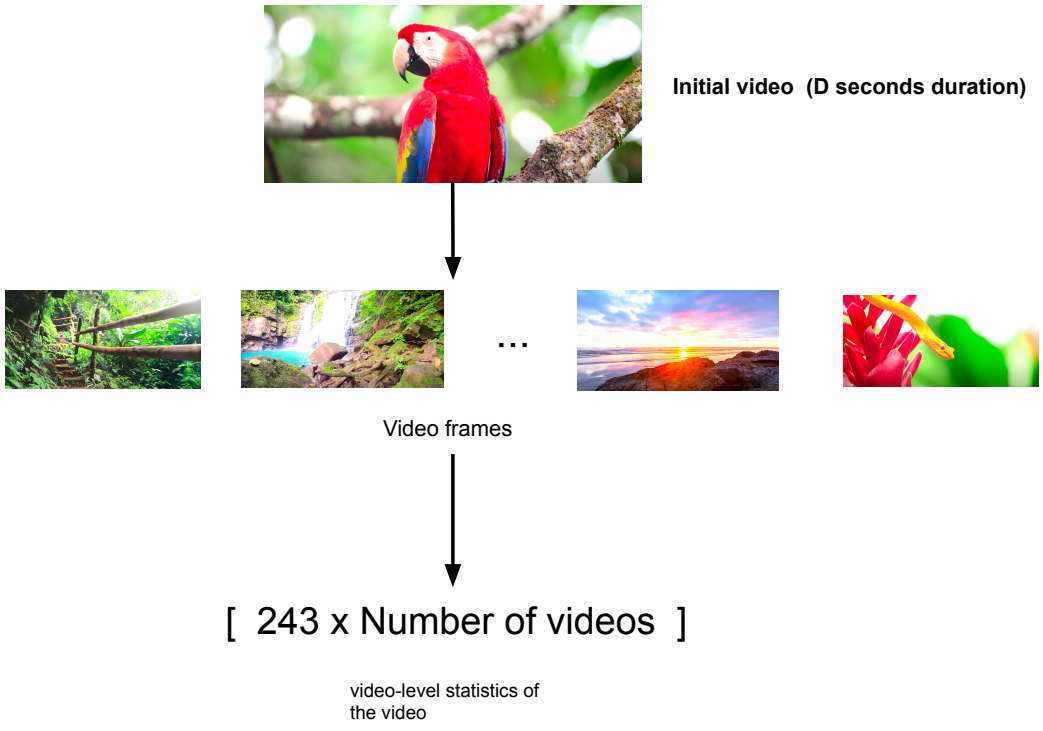


Figure 2.2: Conceptual diagram of the visual feature extraction process

The aforementioned features provide a wide range of low (simple color aggregates), mid (optical flows) and high (existence of objects and faces) representation levels. The rationale behind the selection of this wide range of types of features lies

in the fact that our goal is to cover every type of information that may possibly be correlated to the visual "informativeness" of the video.

Chapter 3

Dataset Description

3.1 Overview

First, before we refer to the implementation of this thesis, we need to illustrate some basic terms associated with it.

Initially, the primary term to explain is the term 'shot'. Shot in filmmaking and video production, according to the author specializing in the history of cinema Robert Sklar "is a series of frames that runs for an uninterrupted period of time".[25] The second basic term we need to explain is the camera movements in a movie. Camera movement in filmmaking is a technique that causes a change in frame or perspective through the movement of the camera.[26] Camera movements are very important in the direction of a film because through these directors can cause many feelings to the audience. For example, fast camera movements can cause the viewer anxiety or even irritation.

In this work, we want to classify the types of shots based on the cinematic aesthetics of camera movements. For this purpose, we needed a trusted dataset. The shots classes of the dataset must be adequately organized, and at the same time, the number of samples must be sufficient. After research, we concluded that there is no corresponding dataset and thus, part of this work became the creation of a new one. In particular, we chose 48 films in which shot detection was applied and then we separated these shots into individual files as described in subchapter 3.2.

In subchapter 3.3, we report all the shots classes that we defined after research. In the last two subchapters, 3.4 and 3.5, we analyze the annotation process where 17 different annotators are asked to designate each shot in which class belongs. Then we analyze the results from all annotations by evaluating the whole process.

3.2 Shot generation

Our primary goal was to be able to recognize in a video when the shots change. The recognition of shot changes was a significant factor for the continuation of the process. We wanted recognition of shots to perform appropriately to create our dataset, so it was essential to make it as precise as possible. Our goal was that the dataset had to contain shots coming from regular movies, where each shot should be organized to the belonging class. The first factor is how big is the difference in the black and white colors of the current frame compared to the previous (gray_diff). The second factor in recognizing the change of shots is the difference in brightness as it is presented in the current frame in relation to the previous (f_diff) with a value of 0 meaning that the image is too dark and 100 that it is very bright. Another factor is the average value of magnitudes of flow vectors (mag_mu). Vector flows are a vector-valued functions $F : R^2 \rightarrow R^2$, e.g., a vector field with two dimensions that can be visualized with a field of arrows. [27] So if the value of magnitudes of flow vectors changed abruptly from the current frame to the next, then it would probably occur a shot change. The last parameter we have taken into consideration is the time of the current shot (current_shot_duration). We had to find the appropriate threshold values for each of the above parameters to decide if there is a shot change. For this purpose we started a hyperparameter tuning process in which we have tested many threshold values for each parameter to find the appropriate. We chose a set of clips from real movies. These clips lasted from 1 minute up to 5:30 minutes.

Initially, to know how accurate the shot detection process is, we had to find the timestamps when the shot is changed, so we annotated each video separately, marking the timestamps. We completed all annotations and tested various threshold values for gray_diff, mag_mu and f_diff. If the parameter, for example, gray_diff, was greater than the value we defined as the threshold, it seems that the difference in

Table 3.1: Values of parameters for hyperparameter tuning

mag_mu	gray_diff	f_diff
0.04	0.22	0.002
0.05	0.55	0.02
0.06	0.75	0.04
0.08	0.95	0.05
0.1	1.5	0.06

black and white was too big and possibly due to the change of shot. With the same mentality, we considered that shot change happened with the rest of the values. Current_shot_duration was not tested with different parameters; we simply set the shot to have at least 1.1 seconds of duration. If two parameters passed the thresholds we had defined, then we assume that changes were steep and shot change happened.

We defined a range of values between all parameters, as seen in table 3.1, and we tested every possible combination. Specifically, we tested five different values for every parameter, so we had 125 different combinations. To find the most accurate one, we set three metrics Precision, Recall, and F1 score. The process we followed to measure the accuracy of each combination was as follows. Since we defined three values for the parameters mentioned above, the shot detection process started in a specific number of videos. Upon completing the shot detection, the return of lists containing the timestamps where the shot changes happened for each movie. The final stage compared the returned timestamps with the annotated timestamps and, depending on the deviation, Precision, Recall, and F1 score was calculated.

Completing the process of Hyperparameter Tuning we plotted the results as shown in Figure 3.1. The x-axis appears as the experiment number, while the percentage score of metrics is in Y-axis. From the plot, we realized which parameters had better results, so we concluded in which price range we had to focus on further analysis. We have noticed that better results appear in cases where mag_mu prices were 0.06 or 0.08, gray_diff 0.22, 0.55, or 0.75, while f_diff values 0.002 or 0.2. Then followed the second phase of hyperparameter tuning, where we tested the values that gave us the best results in the first phase. In table 3.2, the results of the second phase appear in detail, where we notice that the best values for parameters are

Table 3.2: Results of the second phase of hyperparameter tuning

mag_mu	gray_diff	f_diff	Recall	Precision	F1
0.06	0.22	0.002	0.78	0.64	0.70
0.06	0.55	0.002	0.74	0.65	0.69
0.06	0.75	0.002	0.52	0.73	0.60
0.06	0.22	0.02	0.75	0.67	0.70
0.06	0.55	0.02	0.71	0.71	0.71
0.06	0.75	0.02	0.49	0.77	0.59
0.08	**0.22**	**0.02**	**0.77**	**0.71**	**0.73**
0.08	0.55	0.002	0.73	0.72	0.72
0.08	0.75	0.002	0.50	0.75	0.60
0.08	0.22	0.002	0.68	0.73	0.70
0.08	0.55	0.02	0.69	0.69	0.69
0.08	0.75	0.02	0.49	0.76	0.59

mag_mu: 0.08, gray_diff: 0.22 and f_diff: 0.02.

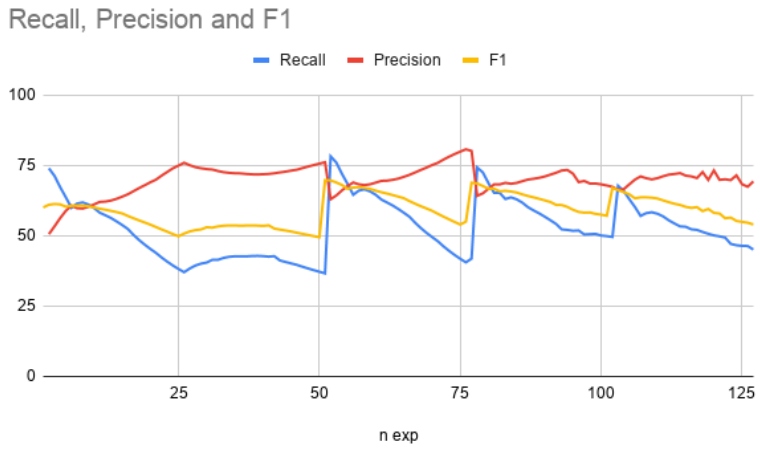


Figure 3.1: Recall, Precision and F1 scores of experiments in hyperparameter tuning process

At this point, we had a trusted way that we could recognize when a shot change occurred. So we created a script that accepts a video file, timestamps where shots change, made video crop and produced new videos. For example, if the shot changed in a video four times, four new videos will be created.

Our next step was to use the above script we created (shot_generator) in multiple videos to produce shot files. So we chose 48 different movies. These films differ from each other to capture different types of shots so that the final dataset to be as

balanced as possible. For example, a film director can be characterized by many static shots and does not include other types and so the number of static shots be can numerous and the rest of it can be reduced. So we have applied a shot generation at 48 films, and about 80 thousand shots were created. Then we deleted all files of a small size, less than 500 Kilobyte, which means that they lasted 1 to 2 seconds, and it would be challenging to annotate by a user in which class belongs. The number of files decreased significantly, but because we wanted to have a manageable video number, we finally kept 4076. In this way, we generated 4076 shots.

3.3 Shot classes

Investigating online and at the same time collaborating with a director, we set the basic types of camera movement we can meet in a shot. For brevity, we call them "shot types". Below are all types of shots, along with similar examples:

1. **Static**, The camera is locked down on a tripod or pedestal and remains still. It is commonly used in dialogues. A static camera does not necessarily dictate a static scene. Actors and even the background can move while the camera remains still. (Link-for-static-video)

2. **Vertical movement**, of the camera lens while the camera remains locked on a tripod. It is like tilting your head up/down. (Link-for-vertical-movement-video)

3. **Tilt**, moving the entire camera up or down without moving its lens. Tilting up is like moving up your entire body from a sitting position. (Link-for-Tilt-video)

4. **Panoramic**, Lateral movement of the camera lens while the camera remains locked down on its tripod or pedestal. It is like moving your head from one side to another. (Link-for-Panoramic-video)

5. **Panoramic Lateral**, The camera follows the action moving parallel to characters. Specifically, the camera captures the lateral movement of the subject.

the camera moves parallel to a person walking down the street to keep them in the frame. (Link-for-Panoramic-lateral-video)

6. **Panoramic 360**, a semicircular movement of the camera. The subject is placed in the center of the frame (usually it doesn't move). The camera moves on tracks. The tracks are curved. (Link-for-Panoramic360-video)

7. **Travelling in**, in this type of shot, the camera moves forward, pushes in a character or follows a character. (Link-for-Travelling-in-video)

8. **Travelling out**, in this type of shot, the camera pulls out. It moves away from the subject and reveals its surrounding. (Link-for-Travelling-out-video)

9. **Zoom in**, In this type of shot, we adjust the camera lens so that the image appears much larger and nearer. (Link-for-Zoom-in-video)

10. **Zoom out**, In this type of shot, the entire image appears much smaller and further away. (Link-for-Zoom-out-video)

11. **Vertigo**, A combination of travelling and zoom. The camera moves backward or forward (travelling) while simultaneously the focal length changes in the opposite direction. For example, while the camera is moving backward, it is simultaneously zooming out. It creates a feeling of uneasiness, paranoia and anxiety. (Link-for-Vertigo-video)

12. **Aerial**, the camera is flown above action. For these shots we use: a helicopter, a drone or a crane. (Link-for-Aerial-video)

13. **Handheld**, the camera is moving throughout the filming set while the camera operator is physically holding it. These camera shots are shaky and create a hectic feeling. (Link-for-Handled-video)

14. **Car Front Windshield**, in this type of shot, the camera is mounted on the front windshield. (Link-for-Car-Front-Windshield-video)

15. **Car Side Mirror**, In this type of shot, the camera is mounted on the side mirror. You can see the driver (and co-driver) from the side. (Link-for-Car-Side-Mirror-video)

After a survey, we ended up in 15 types of shots. Some shot types are more widespread and thus they occur more often during a movie, while other types of shots may not appear at all. For example, the most frequently displayed type shot is undoubtedly the static shot because it is the easiest to capture from a director and more relaxing for the viewer. On the contrary, Vertigo is a particular type and appears in limited kinds of movies.

3.4 Shot annotation

After the video collection process, our goal was to use them for training a set of machine learning models. But in order this to take place, we had to know each video-shot, in which class belongs to create an organized dataset where each directory would contain videos of the same class.

For this purpose, the video annotation process took place. The video annotation process classifies video to a class as ground truth for training and testing the proposed video class. This process was executed through a web application, which was explicitly designed for this purpose. In detail, 17 people asked to see and annotate some videos in order to construct the ground truth. The application was capable of randomly serving all the videos, one by one, to the end-user, while the user was able to watch the whole video, go back and forth in time, and note what type of shot the video is.

Each user was linked to the web application and provided him instructions for the procedure to make annotations. Initially, he had to make a sign-up if he had not been re-connected to the platform or login in case he had made annotation previously. Also, users were given documentation as a guideline that explained any camera movement along with the videos containing examples of each movement. As we see in Figure 3.2, after the user's login, a random video appeared. The user could see the video as many times as he wanted and even go to any point of the video he wishes. As long as the user recognized in which class shot belongs, then he selected the class. At this point, it should be noted that we added a category named N/A. The user chose this class if the video was corrupted or belonged in a category that did not exist in the options. As shown in Figure 3.3, after the user has completed

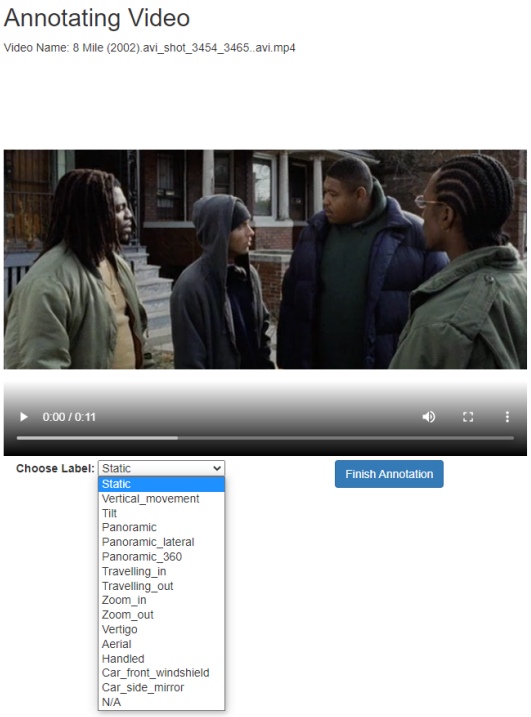


Figure 3.2: Annotation page of web application

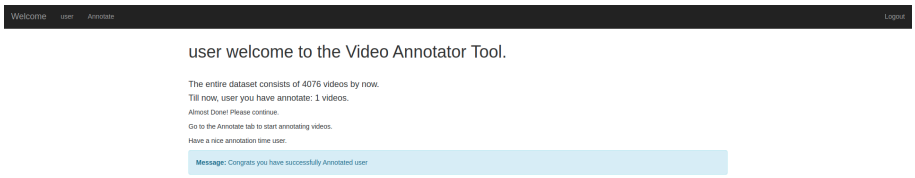


Figure 3.3: User annotation page of web application

the annotation for a video, he was transferred to a page where informed him about the number of total videos available for annotation and the number of videos he has annotated. All annotations result stored in a txt file. Specifically, the file saved the date as well as the time where the annotation occurred. Also saved the name of the username, the video name that was annotated, but also in which class it classified it.

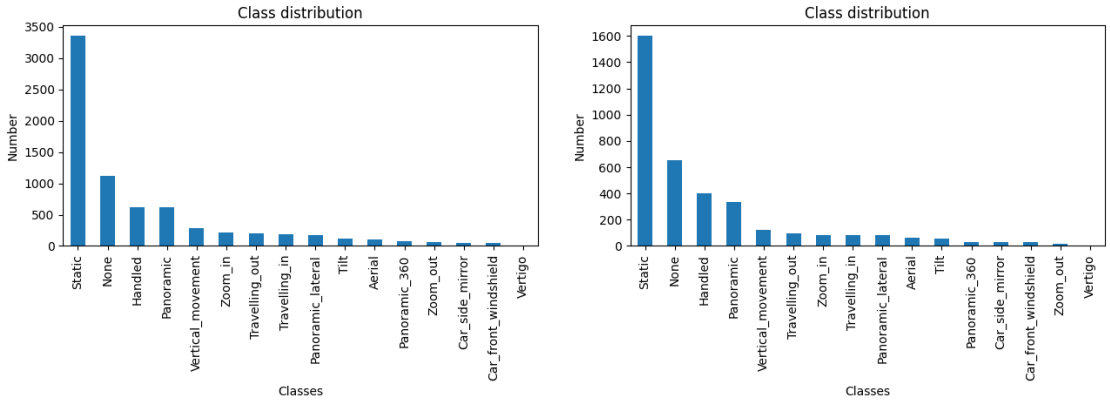
3.5 Annotation data aggregation

Once the annotation process was completed, the individual video classes had to be combined, resulting in an acceptable, final ground truth summary that aggregates all users' opinions who watched and annotated the specific video. This aggregation process is essential for constructing a robust ground truth, since it will be used to evaluate and train the proposed supervised pipeline.

The aggregation process exports valuable pieces of information to have an overview of annotations. In particular, we have extract information such as the number of annotated and no annotated files. Also, the number of annotations made by each user. The resulting video classes that were annotated did not necessarily have the same number of annotators per video, making the construction of a robust dataset more difficult.

The basic information we expected was also the number of samples that exist in each class separately. We have noticed a big difference in samples between the Static class compared to the rest. Users made 3,352 annotations in the Static class, resulting in 44.33% of annotations being Static class. In the second place, follows the annotations that did not belong to any class or the video were corrupted, resulting in None class covering 15.54% of annotations. Then the remaining classes were followed, with the largest one (Handled) having 615 samples while the last one (Vertigo) has only seven samples.

Our next step was to aggregate all users' opinions who watched and annotated the specific video and, depending on the users' agreement, to define each video to what class it belongs. So for each video, taking into account the annotations of users, we ranked it in the appropriate class with the corresponding confidence, e.g., if three users annotate a video, where the two had ranked the video in Static class while one had chosen Panoramic, the winning annotation would be the Static with confidence of 66.66%. So after the above process, the final number of samples at each class became clearer. After the majority, the number of annotated videos was 3693. Analytically the number of samples before and after the user agreement is shown in the figure 3.4.



(a) (b)

Figure 3.4: (a) Number of samples before users' agreement.(b) Number of samples after users' agreement

Then, in order to see in which classes users' agreement was confused, we plotted a confusion matrix. As seen in Figure 3.5, users seem to have a clear picture of each shot category in general. Even categories that resemble each other, such as Panoramic and Panoramic lateral users, did not choose the wrong category. The only class in which there is a slight discrepancy seems to be the Handled, and this because it is a special class. Handled is considered an individual class because it is not an apparent camera motion such as Vertical Movement. Instead, it can be combined with other camera movements, e.g., The shot is being pulled in hand, but at the same time, zoom-in happens. The rate of agreement between users was 89.11%.

The final step was to create the final dataset. Two criteria had to apply to place a shot in a class. The first criterion was at least two annotators must annotate the shot. The second was to have an agreement of annotators greater than 0.6. Based on these two criteria, the final dataset was created consisting of 1.877 videos. Table 3.3 shows that the classes with the corresponding samples containing each.

Summarizing all the aggregation process we observe a heterogeneity of samples in the final dataset. Some types of camera shots are more known, while others are used more rarely. In particular, we observe that static class is a class that appears very often so that the number of samples has a large gap among the other classes. On the other hand, Vertigo is a rare camera motion that e.g. there is a chance not

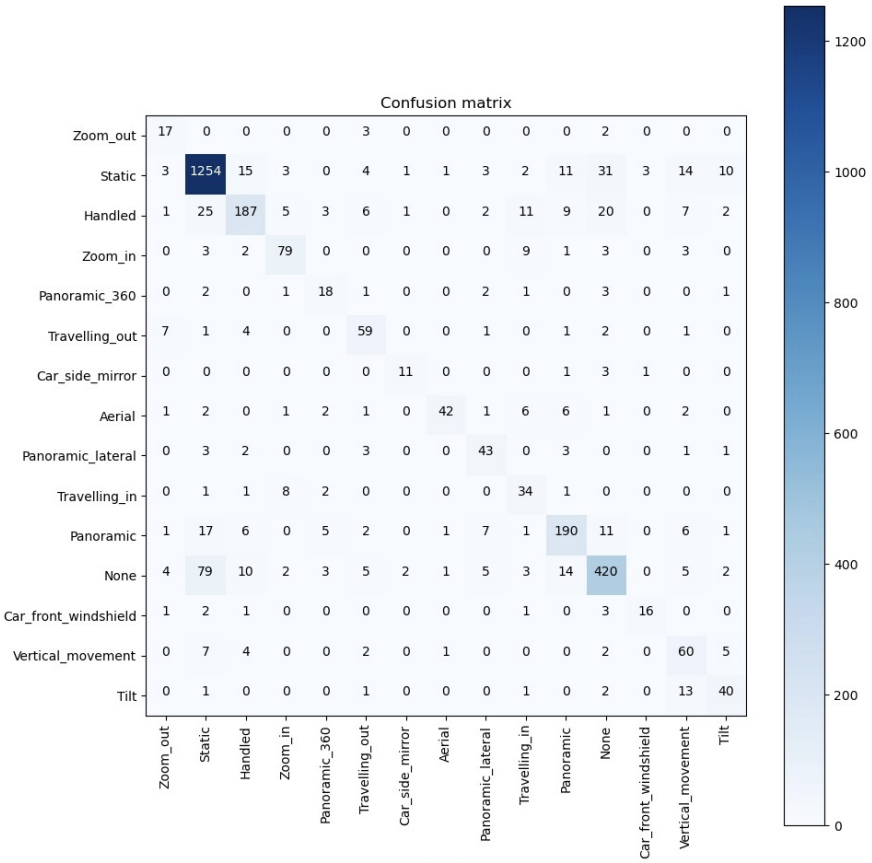


Figure 3.5: Confusion matrix of users' agreement

appear in a whole movie.

Table 3.3: Number of samples per class of the final dataset

Class	Samples
Aerial	51
Static	985
Car front windshield	20
Car side mirror	23
Handled	273
Panoramic	207
Panoramic lateral	46
Panoramic 360	21
Tilt	37
Travelling in	55
Travelling out	46
Vertical movement	52
Vertigo	1
Zoom in	51
Zoom out	9

Chapter 4

Experiments

For the experimental evaluation, we applied three different phases of experiments; each phase's difference was the dissimilar features. Specifically, the first experiments performed using the features that had been extracted, as mentioned in Chapter 2. In each phase of experiments, we faced four different classification problems starting from the most straightforward, i.e., Binary, and gradually adding classes making the problem more complicated, reaching up to 9 classes. The classes we selected were those with the most samples resulting from the users' annotation. In table 4.1, we see the classes that we chose and the number of samples in each of them.

We evaluated and compared the performance of 6 different classification algorithms. We chose different algorithms, namely simple supervised algorithms such as Decision Trees(DT), Support Vector Machines (SVM), and K-Nearest Neighbors (KNN), but also more complex ensembles algorithms such as Random Forest(RF), ExtraTrees(ET), and AdaBoost(AB).

4.1 Shot classification performance

Initially, as mentioned above, we encountered a Binary classification problem in which the model's goal is to distinguish a shot whether it is static or non-static. We chose these two categories because the static class contained the most samples. The number of samples in the static class was 985, while in the non-static class, we put the remaining classes mentioned before, and the number reached 781 samples.

Table 4.1: Number of samples per class

Class	Samples
Static	985
Handled	273
Panoramic	207
Travelling in	55
Vertical movement	52
Aerial	51
Zoom in	51
Travelling out	46
Panoramic lateral	46

Table 4.2: Performance of classifiers in Binary classification task

Classifier	Features	Accuracy	F1 score	Precision	Recall
DT	Original	0.73	0.72	0.72	0.73
DT	Delta	0.74	0.74	0.74	0.74
DT	Delta-Colors	0.75	0.75	0.77	0.77
SVM	Original	0.76	0.75	0.75	0.75
SVM	Delta	0.75	0.74	0.75	0.74
SVM	Delta-Colors	0.75	0.74	0.74	0.74
KNN	Original	0.64	0.63	0.64	0.64
KNN	Delta	0.66	0.66	0.66	0.66
KNN	Delta-Colors	0.67	0.67	0.67	0.67
RF	Original	0.79	0.78	0.78	0.78
RF	Delta	0.78	0.78	0.78	0.78
RF	Delta-Colors	0.77	0.77	0.77	0.77
ET	Original	0.80	0.78	0.79	0.77
ET	Delta	0.80	0.80	0.80	0.80
ET	Delta-Colors	0.79	0.78	0.78	0.78
AB	Original	0.76	0.76	0.76	0.76
AB	Delta	0.78	0.78	0.78	0.78
AB	Delta-Colors	0.76	0.76	0.76	0.76

At this point, it should be noted that the results for each classifier were produced from a group of five, 5-Fold cross-validation iterations. We followed this strategy to increase the most out of data to ensure the best possible outcome, increasing the results' validity. Furthermore, we applied SMOTE[28] to all of our experiments to face off the imbalance of data.

As we can see in table 4.2, the ensembles algorithms had better performance with the original data, reaching up to f1 score up to 0.78. On the contrary, KNN

had the worst performance and by a large margin compared to all other classifiers with an f1 score of 0.63, and at first glance, we assume that KNN may not be the right algorithm we are looking for in our problem.

Afterwards, we extracted additional information from the feature extraction process and added the delta features. Mean of the delta features and std of the delta features are computed for each of the 52 non-object features for the whole video. We added extra features to observe, mainly if additional information helps classifiers to recognize different shots better. After the insertion of Delta information the total number of features arrived up to 348.

Adding the delta information, we noticed that some classifiers' performance improved while in others it decreased. The performance reductions occurred in 1 of the six classifiers. Specifically, the f1 score of SVM was decreased by 1% and we consider the difference negligible. On the other hand, we noticed improvements in other classifiers, where we saw better results. The most significant improvement had KNN, where the f1 score rose to 3% while the remaining one was increased by 2% other than RF, where the result was the same.

Concluding the experiments with the addition of delta information, we hypothesized that some features out of a total of 348 might not help our models to learn better but instead they could make the learning process more challenging. The first features we thought of were all the features related to colors. To make sure that the color features do not contribute positively to the training process, we plotted all the features' histograms to have a picture of their distribution. As shown in figure 4.1 our hypothesis turned out to be correct; all the color features (red, green, blue) had an exponential distribution which means that if perhaps we removed them it will help the models focus better on the necessary features and learn better the essential information.

After noticing the histograms of the color features that had exponential distribution, we decided to remove them to see if there would be an improvement in the classifiers' performance. So we followed a new phase of experiments where we kept the delta features (after noticing an improvement in our classifiers' performance) and removed the colors features. The number of features was reduced from 348 to

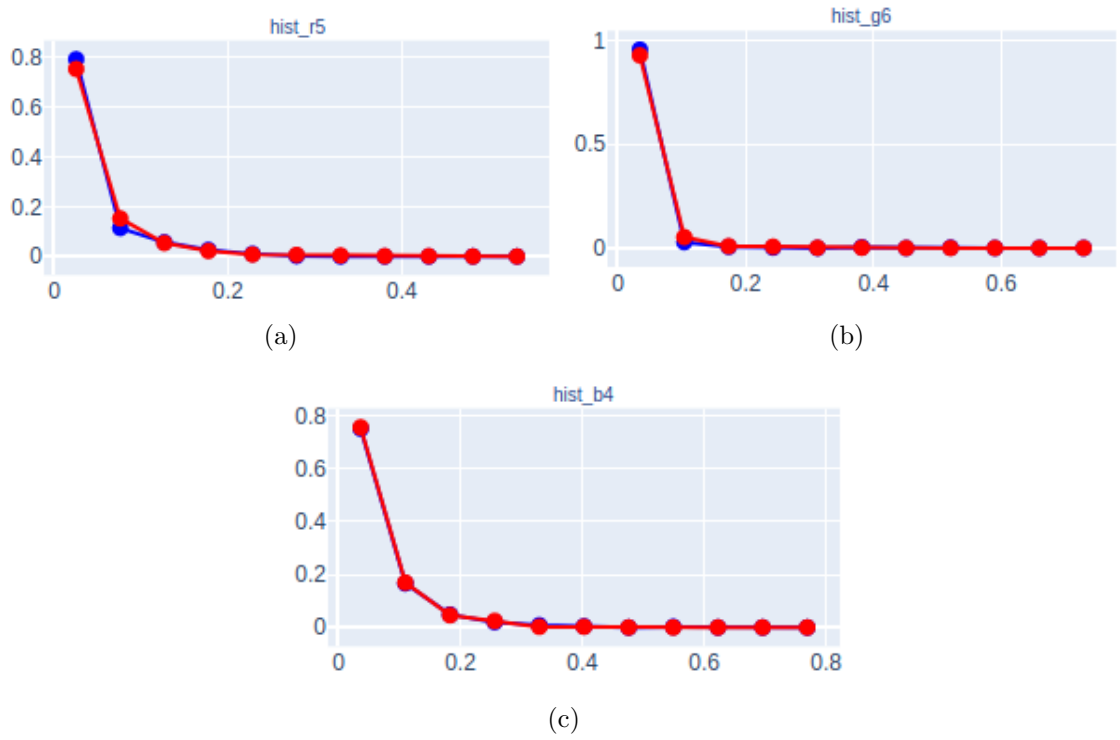
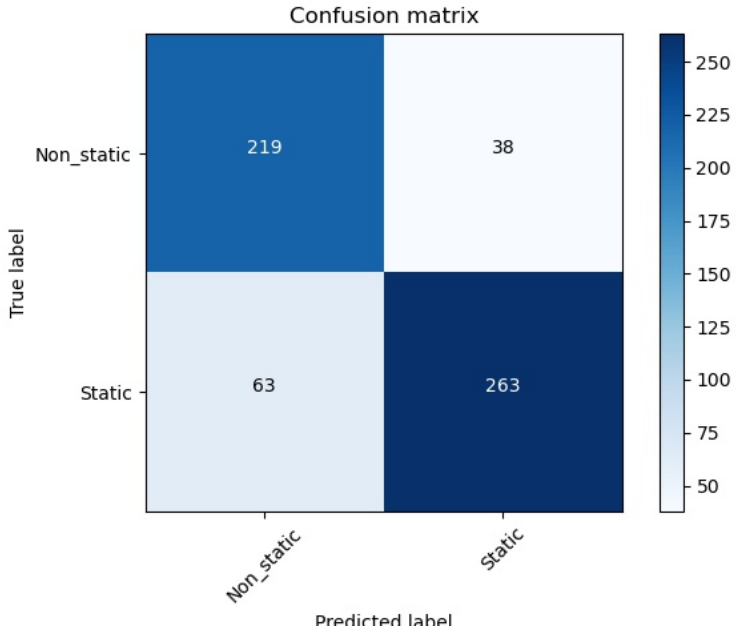


Figure 4.1: (a) Histogram of a red feature between static and non-static classes.(b) Histogram of a green feature between static and non-static classes.(c) Histogram of a blue feature between static and non-static classes

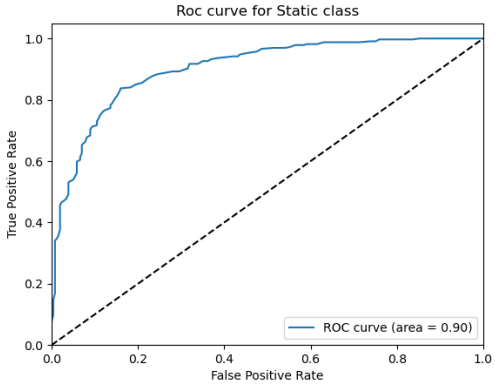
204. We generally observed an improvement to simple supervised classifiers. On the other hand, the performance of ensembles algorithms was decreased by 1% to RF and by 2% to ET and AB.

After performing all the binary classification experiments tasks, we noticed that ET had the best performance of all the classifiers with the delta features without removing the colors. In second place we followed the RF with delta information. The differences between the two best classifiers were 2% to all metrics, with the ET just having a higher accuracy, f1 score minimum precision and minimum Recall 0.80 while the RF had 0.78. Figure 4.2 represents the confusion matrix for the binary classification task and Roc curves for each class separately. In the binary classification task, we observed that the classifier managed to separate the two classes to a satisfactory degree by correctly finding 219 Non-static samples and 263 Static.

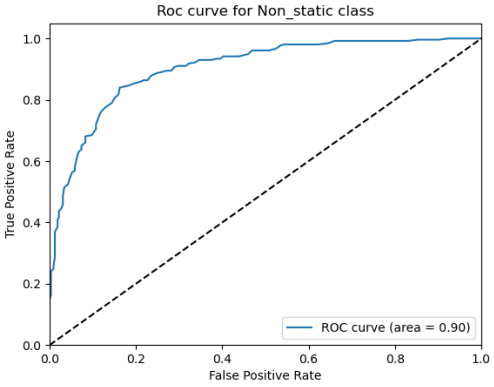
Subsequently, we followed the same experimental technique for tasks with more classes. We gradually add a class to our problem to make it more complex and



(a) Confusion matrix of Binary classification



(b) ROC curve of static class in binary classification



(c) ROC curve of non-static class in binary classification

Figure 4.2: Confusion matrix and ROC curves of every class in the binary classification task.

to observe how robust each classifier is. Initially, we kept Static as the main class because it had enough samples and we created two new classes. The second class we created was Zoom, Zoom contains all the camera movements that the perimeter image during zoom changes at very fast intervals while the central image does not change at all or changes at a slower rate. In this class we added the categories Zoom in, Travelling in, Traveling out with a total number of samples 152. In the third and last class, we added all the vertical and horizontal camera movements which means

Table 4.3: Performance of classifiers in 3 classes classification task

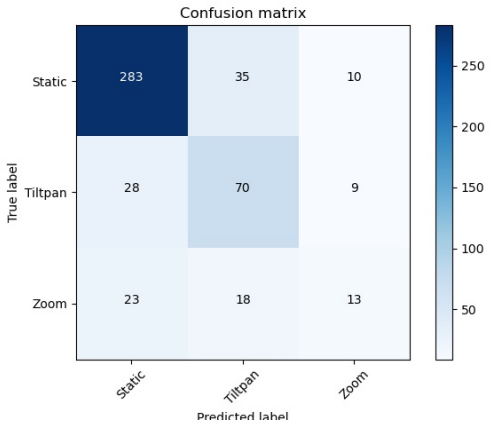
Classifier	Features	Accuracy	F1 score	Precision	Recall	Min Precision	Min Recall
DT	Original	0.52	0.47	0.48	0.48	0.22	0.22
DT	Delta	0.60	0.48	0.48	0.50	0.2	0.32
DT	Delta-Colors	0.62	0.46	0.46	0.48	0.09	0.13
SVM	Original	0.63	0.51	0.51	0.51	0.16	0.12
SVM	Delta	0.67	0.48	0.50	0.47	0.09	0.10
SVM	Delta-Colors	0.71	0.50	0.52	0.49	0.13	0.13
KNN	Original	0.46	0.43	0.45	0.44	0.16	0.37
KNN	Delta	0.48	0.41	0.43	0.44	0.14	0.27
KNN	Delta-Colors	0.51	0.42	0.44	0.45	0.13	0.37
RF	Original	0.64	0.53	0.55	0.54	0.44	0.18
RF	Delta	0.75	0.61	0.63	0.59	0.37	0.24
RF	Delta-Colors	0.73	0.58	0.60	0.58	0.39	0.20
ET	Original	0.67	0.55	0.56	0.56	0.29	0.29
ET	Delta	0.77	0.61	0.66	0.60	0.54	0.25
ET	Delta-Colors	0.75	0.57	0.59	0.56	0.36	0.17
AB	Original	0.64	0.59	0.56	0.57	0.28	0.33
AB	Delta	0.74	0.60	0.59	0.60	0.29	0.34
AB	Delta-Colors	0.70	0.56	0.56	0.58	0.24	0.24

that we placed the categories Panoramic, Panoramic lateral, Vertical Movement, and Tilt with a total number of samples 342.

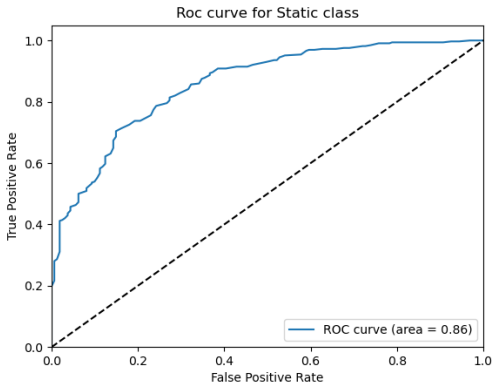
The results from the experiments with the three classes are shown in Table 4.3. The delta information seems to have significantly helped the ensemble algorithms improve their performance, while the removal of colors did not help. Specifically, delta information increased the performance of RF classifiers by 8%, which is impressive. In ET, there was observed an improvement of 6%, while in AB, an improvement of 1%. On the contrary, SVM and KNN did not have good results, as we noticed before in the Binary classification task. These algorithms show a preference for the original data, i.e., without the addition of delta features and the removal of colors. We saw better results in Extra Trees with an f1 score of 0.61, minimum precision of 0.54, and min recall of 0.25.

Observing the confusion matrix in Figure 4.3 we detect a difficulty in ET (which had the best performance) to recognize the Zoom class. It seems that the Zoom movement was an uncertain class since he managed to find correct only 13 out of 54 samples. We see convenience in recognizing the Static class, as in the binary

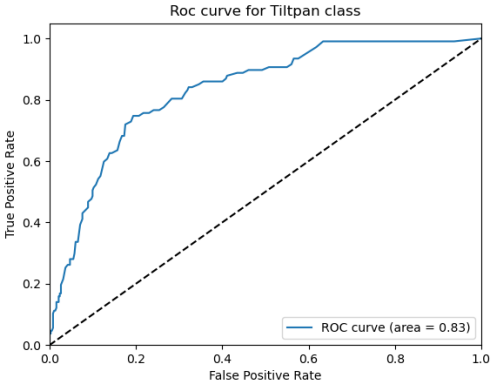
problem, while in Tiltpan, the results are very adequate. At this ponint, it should be mentioned that there was difficulty recognizing the class Zoom from all classifiers and not only by ET. Just for brevity, we do not replicate the confusion matrix of all classifiers.



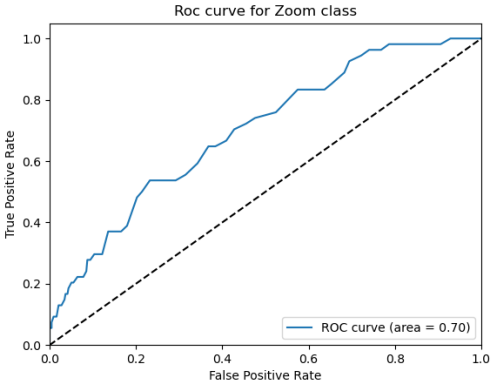
(a) Confusion matrix of 3 classes classification.



(b) ROC curve of static class in 3 classes classification.



(c) ROC curve of Tiltpan class in 3 classes classification



(d) ROC curve of Zoom class in 3 classes classification

Figure 4.3: Confusion matrix and ROC curves of every class in the 3 classes classification task

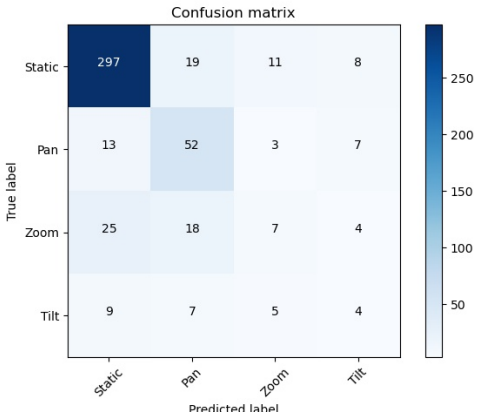
Since the classifiers seemed to learn the camera's vertical and horizontal movements at an adequate level, we divided them into two sub-classes. The camera's horizontal movements (Panoramic, Panoramic lateral) became an individual class while the vertical movements (Vertical movement, Tilt) became the second sub-class. So we started experiments with four classes in total, the classes we had before, namely Static and Zoom, and the new ones Tilt and Pan.

As we see in table 4.4 we observe better results in the case that there are delta fea-

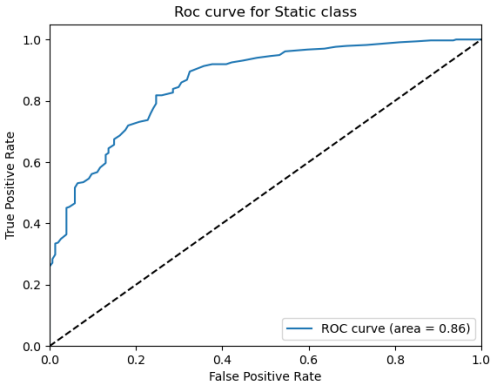
Table 4.4: Performance of classifiers in 4 classes classification task

Classifier	Features	Accuracy	F1 score	Precision	Recall	Min Precision	Min Recall
DT	Original	0.48	0.39	0.40	0.41	0.14	0.29
DT	Delta	0.58	0.40	0.40	0.43	0.13	0.26
DT	Delta-Colors	0.52	0.35	0.36	0.39	0.09	0.25
SVM	Original	0.56	0.38	0.39	0.38	0.17	0.13
SVM	Delta	0.67	0.40	0.42	0.39	0.17	0.14
SVM	Delta-Colors	0.69	0.39	0.39	0.40	0.14	0.093
KNN	Original	0.40	0.32	0.33	0.34	0.10	0.25
KNN	Delta	0.42	0.32	0.34	0.35	0.05	0.19
KNN	Delta-Colors	0.44	0.33	0.35	0.38	0.12	0.26
RF	Original	0.62	0.46	0.47	0.46	0.29	0.20
RF	Delta	0.66	0.38	0.38	0.40	0.05	0.038
RF	Delta-Colors	0.69	0.46	0.48	0.46	0.27	0.21
ET	Original	0.61	0.46	0.47	0.46	0.26	0.20
ET	Delta	0.72	0.44	0.46	0.44	0.17	0.10
ET	Delta-Colors	0.74	0.50	0.53	0.49	0.28	0.16
AB	Original	0.51	0.41	0.42	0.41	0.16	0.18
AB	Delta	0.65	0.41	0.41	0.42	0.15	0.17
AB	Delta-Colors	0.65	0.44	0.44	0.46	0.20	0.14

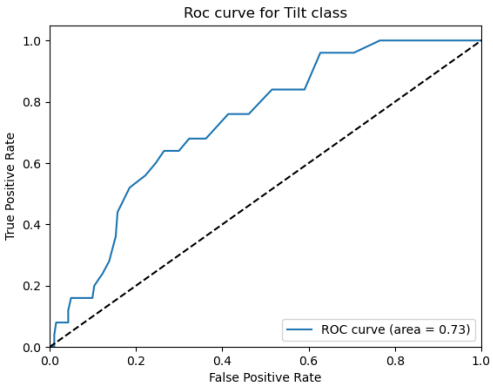
tures without the color information. It seems that reducing the dimensions combined with the increase of the classes, helped most classifiers improve their performance. Mainly, we detect this behavior more intensely in RF, ET, and AB. In contrast to Binary and three-class experiments, the addition of delta features shown to reduce performance compared to the original data means that the volume of features has undoubtedly an essential role. ET had the best results with Delta information without the color features reaching a satisfactory f1 score of 0.50 min precision 0.28 and min recall 0.16. The class that learned at a worse level was Tilt because it contained the fewest samples, i.e., 89.



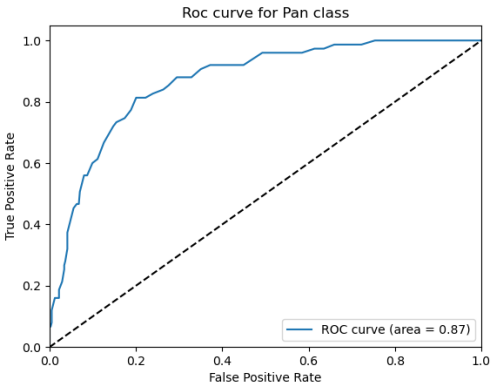
(a) Confusion matrix of 4 classes classification



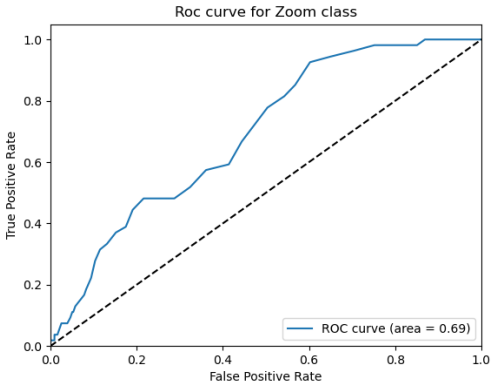
(b) Roc curve of Static class in 4 classes classification



(c) Roc curve of Tilt class in 4 classes classification



(d) Roc curve of Pan class in 4 classes classification



(e) Roc curve of Zoom class in 4 classes classification

Figure 4.4: Confusion matrix and ROC curves of every class in the 4 classes classification task

Table 4.5: Performance of classifiers in 9 classes classification task

Classifier	Features	Accuracy	F1 score	Precision	Recall	Min Precision	Min Recall
DT	Original	0.30	0.16	0.17	0.18	0.0	0.0
DT	Delta	0.41	0.17	0.18	0.18	0.034	0.071
DT	Delta-Colors	0.35	0.16	0.17	0.17	0.0	0.0
SVM	Original	0.40	0.17	0.19	0.18	0.0	0.0
SVM	Delta	0.56	0.19	0.20	0.20	0.0	0.0
SVM	Delta-Colors	0.55	0.21	0.21	0.21	0.0	0.0
KNN	Original	0.21	0.13	0.15	0.14	0.0	0.0
KNN	Delta	0.31	0.16	0.17	0.19	0.05	0.06
KNN	Delta-Colors	0.26	0.12	0.14	0.13	0.0	0.0
RF	Original	0.48	0.22	0.22	0.21	0.0	0.0
RF	Delta	0.60	0.21	0.24	0.22	0.0	0.0
RF	Delta-Colors	0.60	0.23	0.23	0.24	0.0	0.0
ET	Original	0.42	0.20	0.21	0.20	0.0	0.0
ET	Delta	0.59	0.22	0.25	0.22	0.0	0.0
ET	Delta-Colors	0.59	0.21	0.24	0.22	0.0	0.0
AB	Original	0.36	0.24	0.26	0.27	0.0	0.0
AB	Delta	0.20	0.12	0.16	0.14	0.0	0.0
AB	Delta-Colors	0.16	0.10	0.15	0.16	0.0	0.0

After observing classifiers' behavior with more or fewer features and adding more classifiers to our final experiment, we took a big step and faced a 9 class classification problem. The classes we trained as well as the samples of each class, are showing in table 4.5. The first thing we noticed while completing the experiments is that not all classifiers learned certain classes. The classes in which they found it challenging to distinguish were those with the fewest samples, e.g., Traveling out, Zoom in, etc.

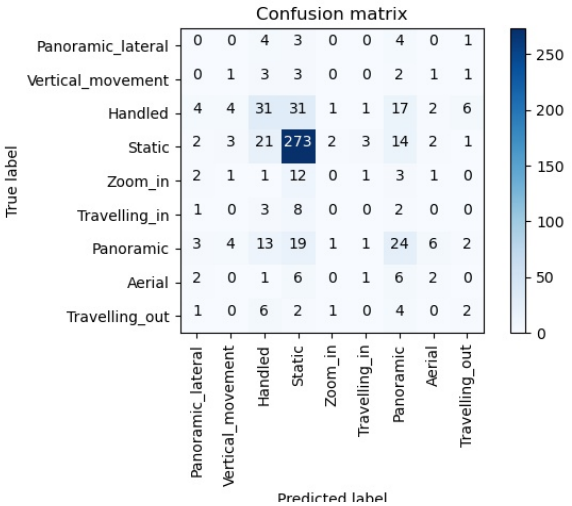
Completing all the experiments, we had to decide the best feature structure and the best classifier. In the experiments, we added up to 3 classes we undoubtedly saw the best results with delta features without removing the colors, so we concluded that most of the information helped the classifiers to separate and learn the classes better. In the last two experiments where the classes increased, we noticed that there where better results in the smaller dimensions, with a reduced number of features. We decided to choose for all our experiments to keep the delta information and the color features for generality reasons. On the classifiers side, ensembles algorithms performed better than other algorithms, especially ET and RF. We chose ET because they had more accurate and balanced results. Table 4.6 shows organized all the ET

Table 4.6: Performance of ET classifier with Delta features

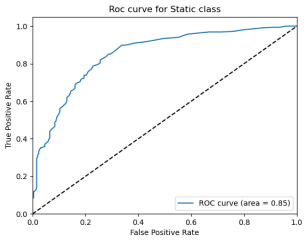
Classifier	Classification	Features	Accuracy	F1 score	Precision	Recall	Min Precision	Min Recall
ET	Binary	Delta	0.80	0.80	0.80	0.80	0.77	0.79
ET	3_Classes	Delta	0.77	0.61	0.66	0.60	0.54	0.25
ET	4_Classes	Delta	0.72	0.44	0.46	0.44	0.17	0.10
ET	9_Classes	Delta	0.59	0.22	0.25	0.22	0.0	0.0

results with the features mentioned above.

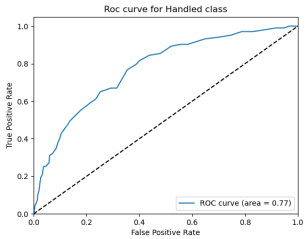
Observing the confusion matrix one after another, we firmly understand that in classes with a small number of samples, the classifier finds it difficult to distinguish them. Also, some classes always have a small percentage of precision and recall in the samples' predictions. Some camera movements are unique and complicated so that the classifier can learn them. For example, the Zoom motion on all the models we trained seemed like a complicated class. We reminded that the Zoom class in the three and four classification problems consisted of the Zoom in, Traveling in and Traveling out classes. Inspecting in more detail the results of all these classes in the 9 class task, in Figure 4.5, we see that the classifier did not learn Zoom in and Travelling in at all. In Travelling out, he found only two samples correctly. Instead, the Static movement was a motion that all classifiers recognized to a large extent, this is because there are a large number of samples in this class, and it is also a camera movement that is simple and easy to detect.



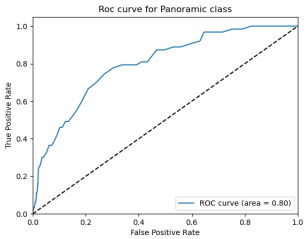
(a) Confusion matrix of 9 classes classification



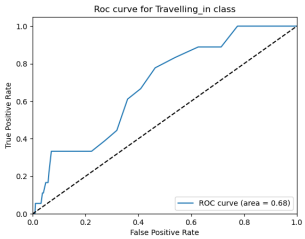
(b) Roc curve of Static class in 9 classes classification



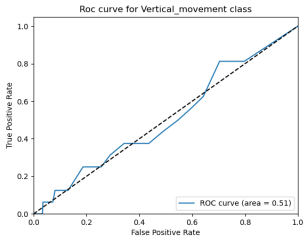
(c) Roc curve of Handled class in 9 classes classification



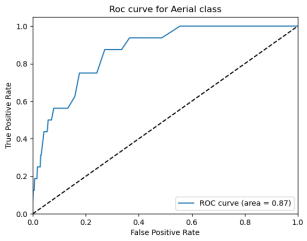
(d) Roc curve of Panoramic class in 9 classes classification



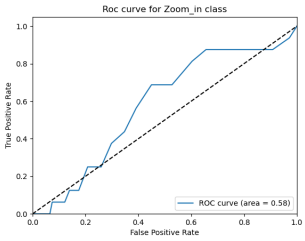
(e) Roc curve of Travelling in class in 9 classes classification



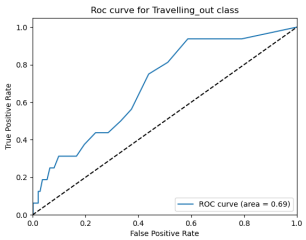
(f) Roc curve of Vertical movement class in 9 classes classification



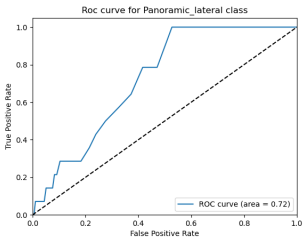
(g) Roc curve of Aerial class in 9 classes classification



(h) Roc curve of Zoom in class in 9 classes classification



(i) Roc curve of Travelling out class in 9 classes classification



(j) Roc curve of Panoramic lateral class in 9 classes classification

Figure 4.5: Confusion matrix and ROC curves of every class in the 9 classes classification task

4.2 Qualitative evaluation and demo

After completing the experimental process, a qualitative assessment of our learned models followed. This process's main objective is to assess how well our trained models can distinguish movies that differ directly.

To this end, we found a set of movies that differ significantly between them directed. Initially, the first set of films we chose were the films named 'Dogma 95'. These films appeared in 1995 by the Lars Von Trier and Thomas Vinterberg director. What made them unique and distinguished was their direction. To consider a movie that belongs to 'Dogma 95', it had to follow ten rules. The primary and most important rule of interest is that the camera must be hand-held. Then there are other rules that follow such as the fact that film must be in color, optical work and filters are forbidden, etc.[29] All 'Dogma 95' movies are 31, and we chose four of them. In particular, we decided on Festen, The Idiots, Julien Donkey-Boy, and Italian For Beginners.

The next set of films we chose were movies directed by Stanley Kubrick. Stanley Kubrick had a unique way of directing his films to stand out mainly by movies of their era. The most important feature of his movies was 'The One Point Perspective', this type appears when it is on the horizon line such as that the objects appear to grow smaller the closer they are to the center.[30] These shots usually lead to camera movements such as traveling in or traveling out where the character moves or walking away to or from the center point, and the camera follows him. Still, a feature of Kubrick's directing is that scenes have slow zoom in or zoom out quite often.[31] We chose four movies, A Clockwork Orange, Barry Lyndon, The Shining, and Eyes Wide Shut.

The four last movies we chose are movies of Steve McQueen. Static shots characterize the inducing director, specifically the long-take static shots. For example, in the movie 'Hunger' there is a shot that lasts 17 minutes and is Static. The films of the inducing director who chose are Hunger, 12 Years A Slave, Widows, and Shame. [32]

Qualitative evaluation results consist of two parts. In the first part, we used

Table 4.7: Predictions of trained models to movies in Binary classification task

		Static	Non-static
Dogma 95	Festen	46.35	53.65
	Italians for beginners	51.08	48.92
	Julien Donkey Boy	46.76	53.24
	The Idiots	38.11	61.8
Stanley Kubrick	A Clockwork Orange	64.67	35.33
	Barry Lyndon	68.80	31.20
	The Shining	59.53	40.47
	Eyes Wide Shut	58.42	41.58
Steve McQueen	12 Years a Slave	56.53	43.47
	Hunger	57.33	42.67
	Shame	51.20	48.80
	Widows	58.37	41.63

the pre-trained models with the best performance, as mentioned in Chapter 4.1, in the movies and make predictions about the types of shots contained and the corresponding percentages of each shot. Using the media vectors of the predictions of each movie we followed a clustering process where films were grouped into clusters.

More specifically, in the first phase, we separated the 12 films into individual shots. It was virtually a shot generation process, as mentioned in Chapter 3.2. Then loaded the pre-trained models and predict each shot separately in which class belongs. To find the final percentage of camera movements in each film, we calculated the average results of all shots. So, according to the classification task returned the rates of the classes contained in each movie. At this point, it should be noted that it became a prediction only at 40% of the shots that became generated for brevity purposes.

Starting from 'Dogma 95' movies, we observe in Table 4.7 that the binary classification task results are very close to what we expected to see. These films are mainly characterized by hand-held camera movement, so we especially observe, in exception to Italians for beginners film, the largest rate belonging to the non-static class. In tables 4.8 and 4.9, we examine in more detail the non-static class. Less often, the vertical and zoom camera moves appear, while the panoramic movements are more frequent. Finally, in table 4.10, where there are nine classes, we see that

Table 4.8: Predictions of trained models to movies in 3 class classification task

		Static	TiltPan	Zoom
Dogma 95	Festen	44.24	34.40	21.36
	Italians for beginners	48.14	30.77	21.09
	Julien Donkey Boy	44.32	35.41	20.27
	The Idiots	34.86	44.27	20.87
Stanley Kubrick	A Clockwork Orange	64.27	21.18	14.54
	Barry Lyndon	69.79	16.42	13.79
	The Shining	59.07	23.79	17.14
	Eyes Wide Shut	61.86	22.56	15.58
Steve McQueen	12 Years a Slave	56.19	28.42	15.38
	Hunger	58.72	24.04	17.24
	Shame	52.08	29.43	18.48
	Widows	57.68	24.34	17.99

Table 4.9: Predictions of trained models to movies in 4 class classification task

		Static	Tilt	Pan	Zoom
Dogma 95	Festen	32.11	18.90	27.69	21.30
	Italians for beginners	37.18	18.60	23.10	21.12
	Julien Donkey Boy	34.39	16.58	25.87	23.16
	The Idiots	25.80	19.61	36.12	19.47
Stanley Kubrick	A Clockwork Orange	56.38	10.14	16.31	17.17
	Barry Lyndon	60.61	10.36	12.28	16.75
	The Shining	49.83	13.08	17.87	19.21
	Eyes Wide Shut	54.28	10.82	16.93	17.97
Steve McQueen	12 Years a Slave	49.98	12.72	21.14	16.16
	Hunger	49.96	13.76	16.60	19.68
	Shame	46.16	12.99	19.76	21.05
	Widows	50.82	12.78	18.08	18.31

the biggest rates belong to static and handled camera movements. As mentioned in Chapter 4.1, the results of the experiments of the 9 class classification task may not be at a satisfactory level, but as in the present example, we can have a general picture for camera movements that exist in a movie. We, therefore, understand from the results of all classes that 'Dogma 95' films are more like non-static camera movements. Most of them belong to horizontal movements and are handle-held.

In Stanley Kubrick's films, the result is not desirable. In these movies, we were waiting to meet camera movements that characterize zoom movement. Instead, observing the results in all classification tasks, the most significant percentage belongs

Table 4.10: Predictions of trained models to movies in 9 class classification task

		Static	Panoramic	Panoramic lateral	Vertical movement	Travelling in	Traveling out	Zoom in	Handled	Aerial
Dogma 95	Festen	22.37	12.25	8.93	7.70	8.29	6.40	5.97	21.93	5.97
	Italians for beginners	28.57	11.46	6.43	7.75	7.34	7.24	7.36	20.57	3.28
	Julien Donkey Boy	22.75	12.89	11.51	6.21	9.09	3.87	5.46	18.39	9.83
	The Idiots	18.38	13.47	13.03	10.88	6.49	6.89	8.02	17.33	5.50
Stanley Kubrick	A Clockwork Orange	44.83	8.87	5.07	5.46	7.13	4.09	6.12	12.67	5.74
	Barry Lyndon	52.59	7.36	3.44	5.58	6.25	4.15	5.54	11.22	3.88
	The Shining	41.42	10.25	6.11	6.24	6.80	4.12	6.18	12.23	6.64
	Eyes Wide Shut	45.56	8.93	3.46	4.98	6.65	5.15	5.82	14.94	4.52
Steve McQueen	12 Years a Slave	40.14	12.39	6.31	7.34	6.98	3.76	5.72	12.30	5.06
	Hunger	37.91	9.51	6.09	6.54	8.79	3.75	6.69	14.17	6.55
	Shame	34.21	11.06	6.48	6.29	7.72	4.67	5.99	16.96	6.62
	Widows	41.20	9.36	5.74	5.80	6.84	4.62	7.04	14.15	5.25

to the static class. We believe they are two factors that have inquired the end result. The first and critical factor is that zoom class is a class that we noticed from the confusions matrix in Figures 4.3, 4.4 and 4.5 that models have difficulty learning to a satisfactory degree. The second factor is that all zoom moves were moving in slow motion, and perhaps the models considered that the camera movement was static. Summarising, we observe that the largest percentage belongs to the static class while the remaining camera movements are shared equally

Finally, we have the movies of Steve McQueen. These films are characterized as mentioned above from log-take static shots. The results in all classification tasks showed an above gradient in the static class. We expected a more significant percentage to belong to static camera motion. Focusing on the binary classification task, where we can better observe the static class, we see that the rates in static class in 3 out of 4 movies are between the range [56%,58%]. In the remaining classification tasks, we observe that there is no big difference between the camera movements. We can not believe that our models did not learn the static class and justify the relatively low rates because the static class had the largest accuracy rate in all classification tasks. We think, however, that the final result was influenced by the fact that was many large static scenes and maybe many small scenes that were not static and thus affected the final average. On the contrary, if many small static scenes characterized the movies, the result was the desired.

Upon completing the first phase, the second followed, where a clustering process was carried out. This procedure was done to realize if, according to the results presented above, we can consider the movies different and conclude that they consist

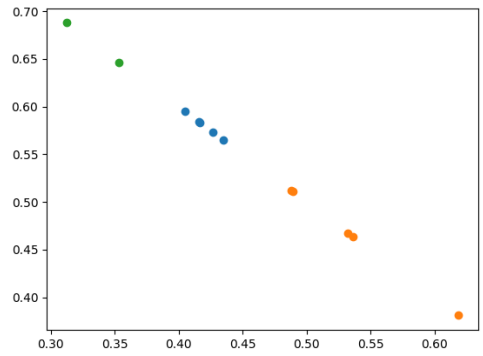
Table 4.11: Clustering predictions

		Binary	3 classes	4 classes	9 classes
Dogma 95	Festen	1	2	1	0
	Italians for beginners	1	2	1	0
	Julien Donkey Boy	1	2	1	0
	The Idiots	1	1	1	0
Stanley Kubrick	A Clockwork Orange	2	0	2	1
	Barry Lyndon	2	0	2	1
	The Shining	0	0	0	2
	Eyes Wide Shut	0	0	2	1
Steve McQueen	12 Years a Slave	0	0	0	2
	Hunger	0	0	0	2
	Shame	1	2	0	2
	Widows	0	0	0	2

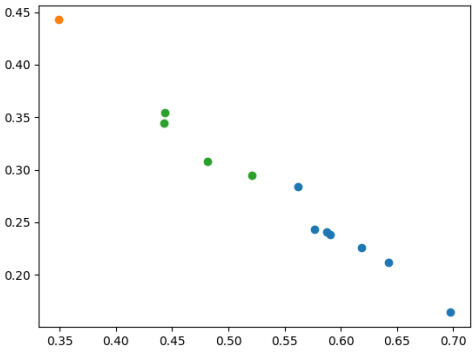
of another director or differ somehow. In particular, in clustering we used the K-Means algorithm. K-Means was accepted as an entrance to the results presented in Tables 4.7, 4.8, 4.9, and 4.10. Thus four different clusterings were made.

Starting with clustering at binary classification as we have seen in Table 4.11 and Figure 4.6, we observe that the 'Dogma 95' films are successfully grouped into the same cluster. The films of Stanley Kubrick's director are grouped to correctly cluster with two out of four movies while the movies of director Steve McQueen with three out of four. In both movie sets, static and non-static rates were close enough, and so K-Means had a hard time separating them into appropriate clusters. Then in 3 classification task was observed K-Mean's difficulty in grouping movies correctly. 'Dogma 95' has managed to result near what we wanted while the remaining two groups of films are grouped as a single cluster. In the remaining two classification tasks, there has been a correct clustering of the movies. 'The Shining' seems to be a movie that looks directed by Steve McQueen's and thus grouped into these films. If we look at it in more detail, the results of 'The Shining' are close enough to the results of Steve McQueen's films, e.g., Static class is at lower levels, and vertical and horizontal camera movements are slightly higher.

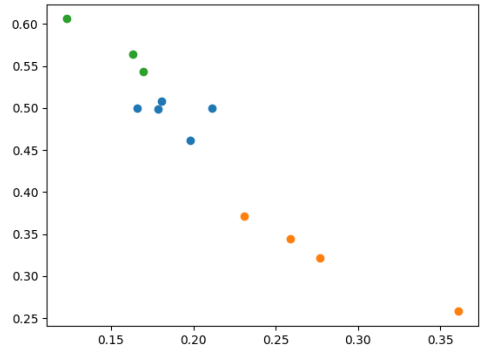
In summary, we would like to remark on the process of qualitative evaluation. From the whole process, we can take conclusions for one or many movies in terms of camera movements made in it/them. Initially, with the binary classification task, we can overview whether the film is more characterized by camera movements that



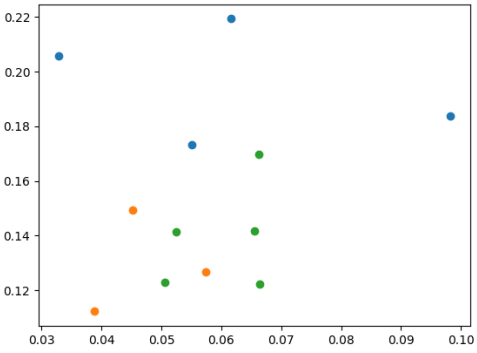
(a) Plots of clusters for Binary classification



(b) Plots of clusters for 3 class classification



(c) Plots of clusters for 4 class classification



(d) Plots of clusters for 9 class classification

Figure 4.6: Plots of clusters for each classification task

are non-static or by static ones. Then with the rest of the classification tasks, we can in greater detail draw more information. We have noticed an increased number of rates (especially more intense to the 9 classes problem) of classes well-learned by the classifier. This means that classes that have been well-learned with confidence can be identified, and we saw it in the 'Dogma 95' movies, where rates of the handled class were much more increased compared to the rest. In the clustering process, we can confidently say that as many classes as the clustering algorithm can better separate the movies. This is because there is further information about the films and can more accurately separate them into clusters.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

In this work, we presented a classification approach to various types of shots we can meet in a movie. Initially, we found a precise way to identify shots during a movie and then save these shots to individual files. Using various annotators to make annotations to about 4 thousand files, we created our novel dataset according to the user agreement. We followed 3 phases of experiments, where all experiments applied to our dataset. Each phase has experienced different features to find which features lead to better results. We also implemented four classification tasks, Binary, 3 Classes, 4 Classes, and 9 Classes. Finally, we created a demo in which we used the Extra Tree classifier that had the best results to observe how well it can separate the classes between them in all classification tasks in real movies. After Demo's completion, we concluded that we could make significant conclusions for a film, such as the aesthetics and the type. Even in the classifier's predictions in the large number of classes where the results were not satisfactory, we observe that the results are correlated with the style and aesthetics of the film.

5.2 Feature work

As future work, it would be interesting to train all classification tasks using deep learning algorithms to observe if our model performance will be better. More

specifically, it would be interesting to use sequence to sequence models, LSTM and Transformers. Furthermore, collect more data and then follow a training process with more classes. Another possible future direction could be to create a database of directors that some directorial features will characterize them. Specifically, a director will be described as using specific camera movements combined with their respective rates. After applying the trained model to a movie, a comparison of the results with the database will be applied. After finishing the comparison process, it will appear to whom or which directors probably the movie belongs.

References

[1] Statista. (2021, January 18). Movie releases in North America from 2000–2020. https://www.statista.com/statistics/187122/movie-releases-in-north-america-since-2001/

[2] Follows, S. (2018, November 21). How many films are released each year? Stephen Follows. https://stephenfollows.com/how-many-films-are-released-each-year/

[3] Statista. (2021b, April 21). Netflix subscribers count worldwide 2013–2021. https://www.statista.com/statistics/250934/quarterly-number-of-netflix-streaming-subscribers-worldwide/

[4] Keys, M. (2021, May 14). Disney Plus nets 103 million global subscribers. The Desk. https://thedesk.matthewkeys.net/2021/05/disney-plus-hulu-espn-plus-103-million-subscribers/

[5] Peter, S. R., N. (2009). Artificial Intelligence: A Modern Approach [ARTIFICIAL INTELLIGENCE 3/E] [Hardcover] (3rd ed.). Prentice Hall.

[6] Mohri, M., Rostamizadeh, A., Talwalkar, A. (2012). Foundations of Machine Learning (Adaptive Computation and Machine Learning series). The MIT Press.

[7] Brownlee, J. (2019, May 22). Difference Between Classification and Regression in Machine Learning. Machine Learning Mastery. https://machinelearningmastery.com/classification-versus-regression-in-machine-learning/

[8] Nadeau, C., Bengio, Y. (2003). Inference for the generalization error. Machine learning, 52(3), 239-281.

[9] Wikipedia contributors. (2021, March 26). Precision and recall. Wikipedia. https://en.wikipedia.org/wiki/Precision_and_recall

[10] Batarseh, F. A., Yang, R. (2020). Data Democracy: At the Nexus of Artificial Intelligence, Software Development, and Knowledge Engineering (1st ed.). Academic Press.

[11] Diez, P. (2018). Smart Wheelchairs and Brain-computer Interfaces. Elsevier Gezondheidszorg.

[12] Classification: ROC Curve and AUC — Machine Learning Crash Course. (2020, February 10). Google Developers. https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc

[13] Quinlan, J. R. (1996, August). Bagging, boosting, and C4. 5. In AAAI/IAAI, Vol. 1 (pp. 725-730).

[14] Vapnik, V., Guyon, I., & Hastie, T. (1995). Support vector machines. Mach. Learn, 20(3), 273-297.

[15] Zhang, M. L., Zhou, Z. H. (2007). ML-KNN: A lazy learning approach to multi-label learning. Pattern recognition, 40(7), 2038-2048.

[16] Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.

[17] Breiman, L. (1996). Bagging predictors. Machine learning, 24(2), 123-140.

[18] Geurts, P., Ernst, D., Wehenkel, L. (2006). Extremely randomized trees. Machine learning, 63(1), 3-42.

[19] Freund, Y., Schapire, R., Abe, N. (1999). A short introduction to boosting. Journal-Japanese Society For Artificial Intelligence, 14(771-780), 1612

[20] AdaBoost Tutorial · Chris McCormick. (2013, December 13). Chris McCormick. http://mccormickml.com/2013/12/13/adaboost-tutorial/

[21] Kurama, V. (2021, April 9). A Guide To Understanding AdaBoost. Paperspace Blog. https://blog.paperspace.com/adaboost-optimizer/

[22] Viola, P., Jones, M. (2001, December). Rapid object detection using a boosted cascade of simple features. In Proceedings of the 2001 IEEE computer society

conference on computer vision and pattern recognition. CVPR 2001 (Vol. 1, pp. I-I). IEEE.

[23] Liu,W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., Berg, A. C. (2016, October).651 Ssd: Single shot multibox detector. In European conference on computer vision (pp. 21-37). 652 Springer, Cham.

[24] Lucas, B. D., Kanade, T. (1981, April). An iterative image registration technique with anapplication to stereo vision.

[25] Sklar, Robert. Film: An International History of the Medium. [London]: Thames and Hudson, [c. 1990]. p. 526.

[26] DeGuzman, K. (2021, February 1). Types of Camera Movements in Film Explained: Definitive Guide. StudioBinder. https://www.studiobinder.com/blog/different-types-of-camera-movements-in-film/

[27] Vector fields as fluid flow. (n.d.). Mathinsight. Retrieved June 14, 2021, from https://mathinsight.org/vector_field_fluid_flow

[28] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. Journal of artificial intelligence research, 16, 321-357.

[29] Dogme 95. (2021, March 21). In Wikipedia. https://en.wikipedia.org/wiki/Dogme_95

[30] Chaubey, V. (2019, July 2). The Shining: The Subtle Ways that Stanley Kubrick Unsettles the Audience. Medium. https://medium.com/the-film-odyssey/the-shining-the-subtle-ways-that-stanley-kubrick-unsettles-the-audience-9982b2af5a2e

[31] Marshall, C. (2012, September 8). Signature Shots from the Films of Stanley Kubrick: One-Point Perspective. Open Culture. https://www.openculture.com/2012/09/signature_shots_from_the_films_of_stanley_kubrick.html

[32] Hogenson, P. (2018, March 2). Immersion into the Work of Steve McQueen. Facets Blog. https://facets.org/blog/exclusive/immersion-into-the-work-of-steve-mcqueen/

References