Universidade do Minho
Escola de Engenharia
Departamento de Informática

Francisco André Vieira Reinolds

Deep Learning for Activity Recognition in Real-Time Video Streams

December 2021

Universidade do Minho
Escola de Engenharia
Departamento de Informática

Francisco André Vieira Reinolds

Deep Learning for Activity Recognition in Real-Time Video Streams

Master dissertation
Integrated Master's in Informatics Engineering

Dissertation supervised by
José Machado

December 2021

ACKNOWLEDGEMENTS

I would like to start by thanking my formal and informal Supervisors for this Thesis, Prof. José Machado, and PhD Candidate Cristiana Neto for all their attention, availability, dedication, guidance, and many more virtues than I know adjectives for. This Thesis wouldn't be complete if it weren't for your invaluable contributions throughout its duration.

To my family that accompanied me and supported me through this stage of my life, thank you to every single one of you, words are not enough to show my appreciation for all the assistance that I received.

To my partner, Normandie, thank you for all the support you gave me from the early stages of the Thesis, through all the challenges, and most recently, with our move to the Netherlands for a new life. For all of these things, and more, thank you.

To my group of friends, that I got to know during these University years, thank you for giving me such a warm welcome in my first year, when I knew no one in a brand new city. All the memories that we made together, will be cherished for the rest of my time, thank you Mansos de MIEI.

STATEMENT OF INTEGRITY

I hereby declare having conducted this academic work with integrity.

I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration.

I further declare that I have fully acknowledged the Code of Ethical Conduct of the University of Minho.

ABSTRACT

In an ever more connected world, smart cities are becoming ever more present in our society. In these smart cities, use cases in which innovations that will benefit its inhabitants are also growing, improving their quality of life. One of these areas is safety, in which Machine Learning (ML) models reveal potential in real-time video-stream analysis in order to determine if violence exists in them.

These ML approaches concern the field of Computer Vision, a field responsible for traducing digital images and videos, and be able to extract knowledge and understandable information from them, in order to be used in diverse contexts. Some of the available alternatives to recognise actions in video streams are based on ML approaches, such as Deep Learning (DL), that grew in popularity in the last years, as it was realised that it had massive potential in several applications that could benefit from having a machine recognising diverse human actions.

In this project, the creation of a ML model that can determine if violence exists in a video-stream is proposed. This model will leverage technology being used in State of the Art methods, such as video classifiers, but also audio classifiers, and Early/Late Fusion (EF / LF) schemes that allow the merging different modalities, in the case of the present work: audio and video. Conclusions will also be drawn as to the accuracy rates of the different types of classifiers, to determine if any other type of classifiers should have more prominence in the State of the Art.

This document begins with an introduction to the work being conducted, in which both the its context, motivation and objectives are explained. Afterwards, the methodology used in order to more efficiently conduct the research in this Thesis is clarified. Following that, the State of the Art concerning ML based approaches to Action Recognition and Violence Detection is explored. After being brought to date in what are the State of the Art approaches, one is able to move forward to the following chapter, in which the Training method that will be employed to train the models that were seen as the best candidates to detect violence is detailed. Subsequently, the selected models are scrutinized in an effort to better understand their architecture, and why they are suited to detect violence. Afterwards, the results achieved by these models are explored, in order to better comprehend how well these performed. Lastly, the conclusions that were reached after conducting this research are stated, and possibilities for expanding this work further are also presented.

The obtained results prove the success and prevalence of video classifiers, and also show the efficacy of models that make use of some kind of fusion.

KEYWORDS Machine Learning, Deep Learning, Action Recognition, Violence Detection, Early Fusion, Late Fusion

RESUMO

Num mundo cada vez mais conetado, as cidades inteligentes tornam-se cada vez mais presentes na nossa sociedade. Nestas cidades inteligentes, crescem também os casos de uso nos quais podem ser aplicadas inovações que beneficiarão os seus habitantes, melhorando a sua qualidade de vida. Uma dessas áreas é a da segurança, na qual modelos de Aprendizagem Máquina (AM) apresentam potencial para analisar streams de vídeo em tempo real e determinar se nestas existe violência.

Estas abordagens de AM são referentes ao campo de Visão por Computador, um campo responsável pela tradução de imagens e vídeos digitais, e pela extração de conhecimento e informação inteligível dos mesmos, de modo a ser utilizada em diversos contextos. Algumas das alternativas disponíveis para reconhecer ações em streams de vídeo são baseados em abordagens de AM, tais como Aprendizagem Profunda (AP), que cresceu em popularidade nos últimos anos, à medida que se tornou claro o massivo potencial que tinha em diversas aplicações, que poderiam beneficiar de ter uma máquina a reconhecer diversas ações humanas.

Neste projeto, é proposta a criação de um modelo de Machine Learning que permita determinar a existência de violência numa stream de vídeo. Este modelo tomará partido de tecnologia utilizada em métodos do Estado da Arte como classificadores de vídeo, mas também de classificadores áudio, e esquemas de Fusão Antecipada / Tardia (FA / FT) que permitem a combinação de várias modalidades de dados, neste caso: áudio e vídeo. Serão tiradas também conclusões sobre as taxas de acerto dos diversos tipos de classificadores, de modo a determinar se algum outro tipo de classificador deveria de ter mais prominência

Este documento começa com uma introdução ao trabalho levado a cabo, em que o seu contexto, motivação, e objetivos são explicados. Seguidamente, a metodologia utilizada de modo a mais eficientemente levar a cabo a pesquisa nesta Tese é clarificada. Após isso, o Estado da Arte no que concerne abordagens baseadas em AM para Reconhecimento de Ações e Deteção de Violência é explorado. Depois de ser atualizado em relação a quais são consideradas abordagens de Estado da Arte, é possível avançar para o capítulo seguinte, onde o método utilisado para treinar os modelos que foram considerados como os melhores candidatos para detetar violência é detalhado. Subsequentemente, os modelos selecionados são escrutinizados de modo a melhor entender a sua arquitetura, e porque são adequados para detetar violência. Depois, os resultados conseguidos por estes modelos são explorados, de modo a melhor compreender o desempenho conseguido. Finalmente, as conclusões que foram chegadas a são apresentadas, tais como possibilidades para expandir e melhorar esta pesquisa.

Os resultados obtidos comprovam o sucesso e a prevalência dos classificadores de vídeo, e mostram também a eficácia dos modelos que tomam partido de algum tipo de fusão.

PALAVRAS-CHAVE Aprendizagem Máquina, Aprendizagem Profunda, Reconhecimento de Ações, Deteção de Violência, Fusão Antecipada, Fusão Tardia

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AMV Acceleration Motion Vector. 13
AMV Acceleration Measure Vector. 14
ANN Artificial Neural Network. 10, 15

BoW Bag of Words. 15
BS Batch Size. 22, 34, 35, 36, 37, 38, 39, 40, 41, 43

CM Confusion Matrix. 17, 33
CNN Convolutional Neural Network. 15, 16, 28

DCT Discrete Cosine Transformation. 25
DL Deep Learning. 13, 15, 16
DNN Deep Neural Network. 15
DS Design Science. 6
DSR Design Science Research. 5, 6

EF Early Fusion. 12, 18, 19, 30, 31, 34, 37, 38, 39, 40, 42

IFV Improved Fisher Vector. 15
IS Information Systems. 6
IT Information Technology. 6

KNN K Nearest Neighbours. 14

LF Late Fusion. 12, 18, 19, 24, 31, 32, 34, 39, 41
LR Learning Rate. 35, 36, 37, 38, 39, 40, 41, 43
LSTM Long-Short Term Memory. 15, 16

MC Mixed Convolution. 28
MFCC Mel Frequency Cepstral Coefficient. 18, 19, 25, 30
ML Machine Learning. 3, 4, 5, 6, 9, 11, 12, 13, 14, 15, 17, 20, 21, 24, 31, 43, 44

OViF Oriented Violent Flows. 16

RMV Region Motion Vector. 14
RNN Recurrent Neural Network. 15

STFT Short Term Fourier Transformation. 25

SVM Support Vector Machine. 10, 13, 14, 15

TRoF Temporal-Robust Features. 15

UM Minho University. 3

VDT Violence Detection. 13

ViF Violent Flows. 16

LIST OF TERMS

Requirements Elicitation the act of drawing out or bringing forth requirements . 4, 42, 43

State of the Art the latest and most sophisticated or advanced stage of a technology, art, or science. . 4, 9, 12, 13, 14, 16, 42, 44

1

INTRODUCTION

The present document describes the research and analysis for the development and exploration of a Machine Learning (ML) model that detects violence in a Real-Time Video Stream. The project arose within the scope of the Master's Dissertation of the Integrated Master's in Informatics Engineering at Minho University (UM). This introductory chapter is divided in three sections. A brief contextualisation of this project is presented, including references to key topics of discussion, as well as the main motivation that led to it's realisation as seen in (Section 1.1). The next chapter includes the main objectives proposed for the development of this master's Thesis, whilst offering a description of the performed work (Section 1.2). Lastly, the introduction is concluded with a presentation of the document's structure in order to simplify its reading (Section 1.3).

1.1 CONTEXTUALISATION & MOTIVATION

Amid the constant rise of computational power and ML knowledge, possible fields where they can be put into use are ever increasing. These fields are generally ones where predictions need to be made, and a computer can learn how to make them, based on previously collected data, from which it gains knowledge, and can then act on it in order to make decisions based on data which it has not yet seen. One of them, focuses on the security of the general populace, more precisely, by employing ML techniques in order to detect certain actions that one wishes to identify in a Video Stream. In this regard, one wishes to spot violent acts in everyday scenarios, e.g. a street, a vehicle, among others. With the swift detection of these actions, the notification of the appropriate authorities can be instantaneous, which will contribute to a reduction of inflicted damages, whether they are done to private property, humans, and even animals, as was displayed in such research as (Li et al., 2019; Ullah et al., 2019; Accattoli et al., 2020).

As often is the case in ML, its use in this instance brings various benefits, due to the impracticability of using humans. The employment of humans to monitor video stream for several hours is not feasible, whether it be due to a financial cost, as well as the human cost, as it is unreasonable to assume that a human can spend 8 hours constantly observing a monitor searching for violence, as the person in question would eventually be distracted resulting in non-detection for a certain amount of time.

The utilisation of ML models in order to detect actions is not a novelty, it's been done as soon as 2005, in Dollár et al. (2005), and since then, progress has been made in several regards, from the models itself, to the way of detecting the aforementioned actions.

As of recently, State of the Art articles generally all have excellent accuracy rates, rarely dipping below the 90%.

1.2 OBJECTIVES

As aforesaid, the main objective of this Thesis, is the study and development of a ML model which is able to accurately detect violent scenes, in a Real-Time Video Stream.

Therefore, in the context of this dissertation, the following *Research Questions* arose:

- **Research Question nr. 1:** What methodologies exist and are now being leveraged in an effort to detect violence in video streams?

To answer this question, the following objectives were highlighted:

  — Analysis of the existing methodologies and subsequent choice of the one that best suits the problem at hand;

  — Requirements Elicitation, so as to know what limitations this work presents;

  — Retrieval of scientific articles that feature State of the Art methods, in an effort to learn how the problems identified can be overcomed;

- **Research Question nr. 2:** Which data should be used to train the model, and what characteristics should it possess?

To answer this question, the following objectives related to the collection and setup of data in order to train the model were highlighted:

  — Requirements Elicitation, in order to understand what characteristics the dataset should possess so as to better train the model;

  — Explore existing datasets, disposing of those that do not bear the aforementioned attributes;

  — Setup the chosen dataset, so that it can be used to train and test the model;

- **Research Question nr. 3:** How should the model be developed so that it attains the expected accuracy rates?

To answer this question, the following objectives related to the development of a ML model that is able do detect violence in a video stream were highlighted:

  — Requirements Elicitation, so as to know what objectives the model should accomplish;

— Analysis of current technologies, in order to decide which best suit the established development needs;

Having completed this project, the final model would be able to be applied in a wide variety of scenarios, ranging from being fed data from a security company and notifying its employees to some anomalous activity, a nightclub, or even in a public scenario. This would benefit the community as a whole, as it would add a layer of security to the general population.

1.3 DOCUMENT STRUCTURE

The document is structured as follows:

1. **Chapter I - Introduction:** In Chapter 1, a contextualisation is given as to what motivated the work, its main objectives, and the document's structure;
2. **Chapter II - Research Methodology** In Chapter 2, the Design Science Research (DSR) investigation methodology is presented, in regards to the development of every study case;
3. **Chapter III - State of the Art:** In Chapter 3, theoretical and scientific concepts are presented and documented. It starts with a ML overview, what some of its most relevant paradigms and tasks are. After said overview, a case is made as to why its advantageous to leverage ML in a Violence Detection context, and how it has been done before;
4. **Chapter IV - Training Methodology**: In Chapter 4, an overview of the used training methodology will be had, with special focus on the used pipeline, and explaining why it was designed in that way;
5. **Chapter V - Models** In Chapter 5, all the models will be listed, have its architecture detailed, and the reasoning behind the decision that led to them being chosen explained;
6. **Chapter VI - Result and Discussion** In Chapter 6, the results are analysed, and discussed, taking into account the type of classifier, and the expectations previously set out for them;
7. **Chapter VII - Conclusions & Future Work:** In Chapter 7, reached conclusions are exposed, and what will / can be done in future work is also disclosed;

2

RESEARCH METHODOLOGY - DESIGN SCIENCE RESEARCH

Any research project should adopt methodologies in an effort to rigorously define its phases, processes and methods. Thus, any study must carefully research and analyse multiple methodologies and technologies that may assist the project's viability. Any final decision should be made over what advantages and shortcomings a certain method offers, weighing them, and make a final determination based on that.

Henceforth, this dissertation's development will be follow the DSR methodology, often used on the construction and evaluation of reliable and robust Information Technology (IT) solutions.

In order to successfully conduct this work, several technologies will be leveraged. The main technology to conduct this study will be Python, an open-source programming language that is used in a wide context of applications, which features one of the most complete sets of libraries used to develop ML models. Notwithstanding, during the development phase, the technologies can be subject to change, as some requirements may be altered, resulting in the addition/removal of one or more items.

Several methodologies were thought of to assist Information Systems (IS) investigators in more consistently conducting their research have been proposed, e.g. in Kuechler and Vaishnavi (2008). These attempted to offer a paradigm that would aid them in the creation of applicable, yet rigorous research. One of these was the DSR methodology.

In Peffers et al. (2007), researchers proposed their version of the DSR methodology. They set the objective that the model that they would propose would:

1. be consistent with design science processes in disciplines other than IS;
2. provide a nominal process for conducting the research;
3. provide a mental model for what Design Science (DS) output looks like;

The proposed methodology is structured in a nominally sequential order, however there is no expectation that the researcher(s) following it would always proceed in a sequential order. Instead, they may actually start at almost any step and move outward. The developed method is comprised of six stages, which are listed and detailed below:

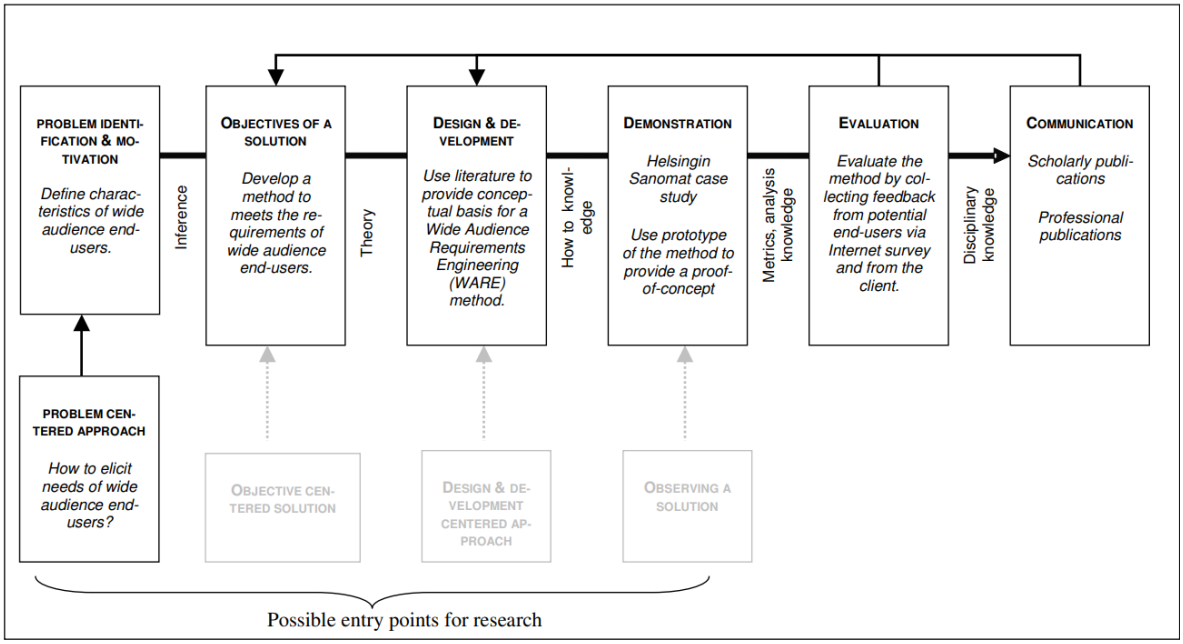


Figure 1: Design Science Research Methodology as proposed by Peffers et al.

2.1 PROBLEM IDENTIFICATION AND MOTIVATIONS

In this phase, the specific research problem should be defined and the value of a solution justified. The authors Peffers et al. indicate that it might be useful to atomize the problems, so that the solution can capture the problem's complexity. In regards to justifying the solution's value, they argue that it not only motivates the research's audience and researchers to pursue the solution and accept the results, it also helps in understanding the reasoning associated with the researcher's understanding of the problem (Peffers et al., 2007).

2.2 OBJECTIVES OF A SOLUTION

In this phase, the objectives of a solution should be inferred from the problem's definition. These objectives can be quantitative, describing for example how long the solution should take in order to solve the problem, or qualitative, e.g., where a new artifact is expected to support solutions to problems no hitherto addressed (Peffers et al., 2007).

2.3 DESIGN AND DEVELOPMENT

In this phase, the artifactual solution is created. This artifact can generally be broadly defined, taking several forms, from an executable file, constructs, models, methods, or instantiations. Most likely, this step of the process will be the most time consuming one, as it encompasses what one could argue are the most meaningful compo-

nents of the whole process. This stage also encompasses the determination of the artifact's desired functionality, its architecture, and the creation of the actual artifact (Peffers et al., 2007).

2.4 DEMONSTRATION

In this phase, the efficacy of the solution should be displayed, which can be achieved via experimentation, simulation, case studies, proof, or other appropriate activities. The results achieved after following this step should be able to, after being analysed, demonstrate the efficacy of the created solution. Resources required for the demonstration include effective knowledge on how to use the artifact to solve the problem (Peffers et al., 2007).

2.5 EVALUATION

In this phase, observations and measures should be made and based on those, determinations on how well the artifact solved the problems should also be made. This stage requires knowledge of relevant metrics and analysis techniques. at the end of this activity, researchers can decide on whether they should iterate back to the 3rd step, in an attempt to improve the effectiveness of the artifact, or to proceed to the next stage (Peffers et al., 2007).

2.6 COMMUNICATION

In this stage, the problem and its importance, the artifact, its utility, efficacy, novelty, rigor of its design and efficacy should be communicated to researchers, and other relevant audiences, such as practicing professionals. In scholarly research publications, researchers might use the structure of this process to structure the paper, just as the nominal structure of an empirical research process (problem definition, literature review, hypothesis development, data collection, analysis, results, discussion, and conclusion) is a common structure for empirical research papers. Communication requires knowledge of the disciplinary culture. (Peffers et al., 2007).

3

STATE OF THE ART

In this chapter, papers and articles featuring State of the Art techniques will be analysed in order to better understand how they are used to overcome the proposed challenges.

An introduction to ML is given, paying special focus to its uses in *Action Detection / Recognition*, whilst explaining where they are better suited to be used, thanks to their perks and downfalls.

3.1 MACHINE LEARNING

ML focuses on the study of how computer algorithms can learn how to make predictions from new data, which has not yet encountered, by training a model on sample data, from which it learns. It does so, by continually updating weights on every node in its model, resulting in a continuous improvement, so that it more accurately predicts the outcome of a certain scenario, based on the data that it has been inputted Mitchell et al. (1997).

By doing so, said algorithms can be trained in an effort to assist humans with several tasks, that would normally require a person to be present, due to a computer's inability to perform them. Such tasks may include, phishing verification, character recognition, speech recognition, among others that require a degree of understanding that a computer was not able to attain Mohri et al. (2018). The coding of a algorithm meant to solve any of these assignments, would be extremely complex, unsuited for any human to do, as it is highly laborious.

Furthermore, the usefulness of ML models not only reveals itself in the above shown examples, where it would be arduous to code an algorithm that would solve the aforementioned problems, its usefulness is also shown when it comes to analysing large amounts of data. Such data is only suited to be examined by a computer, as it would not be feasible for a human to do so, due to its immense nature.

Depending on the nature of the data that needs to be assessed, there are several ML paradigms that can be used to train a model in order for it to better improve itself, and to correctly predict the outcome, for a given data.

3.1.1 *Machine Learning Paradigms*

Supervised Learning

The ML model is trained with labeled data, from which it is able to draw a relation between the input data and the output result.

This paradigm is often used in classification, regression and ranking tasks Mohri et al. (2018). The most widely used supervised learning algorithms are Support Vector Machine (SVM), Linear Regression, Logistic Regression, Naive Bayes, and Artificial Neural Networks (ANNs) (Hoffman and Bhattacharya, 2016).

Unsupervised Learning

In unsupervised learning, the data is given without any labels, and the model looks for previously unnoticed commonalities in the data, reacting on the presence or absence of said commonalities. It contrasts heavily with Supervised Learning, as very little human intervention is required, namely in the data pre-processing stage.

There are several approaches, each of them with several sub approaches, among which: Clustering, Anomaly Detection, ANNs, and Latent Variable models (Siadati, 2018).

Semi-Supervised Learning

Semi-Supervised learning is an hybrid of Supervised and Unsupervised learning, as these are trained on a combination of both labeled and unlabeled data.

This is an advantage in the regards that, one does not need to allocate as much time and effort as would otherwise need, when it comes to labelling the data in order for it to be used. Furthermore, the usage of label may introduce human bias to the model, which is obviously not desirable (Castle, 2018; Mohri et al., 2018).

Active Learning

When actively learning, the learner either adaptively or interactively collects training examples, often by requesting labels for new data points.

The objective is then, to achieve a performance that is comparable to a supervised learning scenario, with fewer labelled examples.

This paradigm is oftentimes used in applications where labelling is costly (Mohri et al., 2018).

Reinforcement Learning

In Reinforcement Learning, the training and testing phases are intermixed, as a consequence of that, the learner actively interacts with its environment, and in some cases affecting it, receiving an immediate reward for each action.

The learner's objective is to maximize its reward over a course of actions and iterations. However, no long-term feedback is provided by the environment, resulting in the learner being faced with the *exploration* versus *exploitation* dilemma (dilemma where one must choose between exploring the unknown in order to gain new information or exploit the already collected information) (Mohri et al., 2018).

3.1.2 *Machine Learning Tasks*

Classification

Classification is the act of assigning a certain category or label to an item.

Classification is used in such problems as image classification, where a model must classify an image, one of the most popular examples of this being the **Cat vs Dog Classification** in which the model must figure out if the animal in an image is a cat or a dog (Mohri et al., 2018).

Regression

Regression is the problem of predicting a *real value* for each item.

Regression can be used in such problems as the prediction of stock values or variations of economic variables. However, in regression, the penalty for an incorrect prediction depends on the magnitude of the difference between the true and predicted values (Mohri et al., 2018).

Pattern Recognition

One of the most common use cases for ML is the detection and recognition of patterns, associations or correlations within data.

Such detection is useful, for example, when it comes to the analysis of bought items and the recommendation of additional items, based on historical analysis, e.g. Jia et al. (2017).

3.1.3 *Machine Learning Models*

There are several ML models each with small variants, even within their general type. As a result of those variants, they naturally have different use cases, stemming from the advantages and disadvantages that are characteristic of them (Mohri et al., 2018).

The choice of which ML model should be used, is one of the most important stages of the development phase and should not be taken lightly, as choosing a model that is unsuitable to solve the desired problem, will more than likely result in a poor performance, and thus, wasted resources.

Some of the most widely used ML models are:

1. **Support Vector Machines**;
2. **Decision Tree**;
3. **Artificial Neural Networks**;
4. **Bayesian Networks**;

3.1.4 *Fusion Schemes*

Semantic indexing in video is perceived as a pattern recognition problem, where given a certain pattern *x*, part of a short *i*, one aims to identify whether a certain semantic context *w* is present in said *i* shot. In order to obtain the pattern representation, one relies on *feature extraction*. Early Fusion (EF) and Late Fusion (LF) differ on the way they integrate the results from feature extraction on the various modalities (Snoek et al., 2005). Fusion schemes, especially audio-video fusion are not widely used in State of the Art articles, however, some research has been done into its theoretical use. Such an article is Song et al. (2019), where the authors review the current status of audio-video fusion and reflect on its features, advantages, disadvantages, use cases, and do some other considerations as to how it should best be used. Another such example is Baltrušaitis et al. (2018), where the authors go more in depth, and indentify broader challenges that multimodal ML faces, such as representation, translation, alignment, fusion, and co-learning. The authors look into each of these problems, describing them, when these occur, and what steps can be taken to mitigate them.

Early Fusion

In EF, indexing approaches first extract unimodal features. After analysing the various unimodal stream, the extracted features are combined into a single representation. in Worring et al. (2007), the authors concatenated unimodal feature vectors in an effort to obtain a fused multimedia representation.

Some of the advantages that stem from using EF, are that it yields a truly multimedia feature representation as the features are integrated from the start, and that since it's a single representation, a single learning phase is needed.

However, it is difficult to combine features into a common representation (Snoek et al., 2005).

Late Fusion

LF begins as well with the extraction of unimodal features, but contrarily to EF, these are not combined so as to allow a model to learn. Prevalently, in LF schemes, the model learns from the extracted and uncombined features, generating scores for each of them, scores which are then combined in order to yield a final score.

From this, it's clear to see that LF schemes focus on the individual strengths of the modalities. With this, comes a significant disadvantage in the expensiveness that comes with having every modality have its own separate supervised learning stage (Snoek et al., 2005).

3.2 MACHINE LEARNING ON VIOLENCE DETECTION

As has been previously approached in this document, ML has ample applications in every sort of field, ranging from Agriculture, Biology, Economy, to mention a few. These applications also vary in their goals, but the most common use cases are the prediction of values, classification of images, etc.

Nonetheless, this dissertation focuses on detecting *Violence Detection* resorting to ML techniques, which has been studied for quite some time. It comes as no surprise that having a computer monitoring a video feed looking for violence in it, has the potential to bring significant advantages, for example in the human resources that would be spared by having a computer do that work.

Violence detection started being investigated after articles like Kuno et al. (1996) were published, in which possibilities to automatically detect humans on video feeds were discussed. One of the first articles that were published in the context of violence detection was Datta et al. (2002). In this paper, the investigators made use of an Acceleration Motion Vector (AMV), that was composed by magnitude and direction of the motion, and used the temporal derivative of AMV in order to define a jerk movement that would then be or not classified as violent. Since then, the State of the Art has of course changed, and the current methods which are used to detect violence have evolved into highly accurate classifiers, most of the articles yielding models that average 85+% prediction rate.

In Ramzan et al. (2019), researchers reviewed State of the Art violence detection techniques, as a way of presenting an exhaustive systematic literature review of the different violence detection methods. They first started by retrieving articles from some of the most reputable scientific publishing websites, like IEEE Explore and Google Scholar, by using prefixes that referred to violence detection. Among the used prefixes were violen*, that with a postfix could result in violent, violence, and detec* that with a postfix could result in detection, detect, detected, etc.

Having collected 2853 articles via an automated search based on the aforementioned strings, they further restricted the articles to 120, after having executed searches based on the article's title. They then read the articles' abstracts which resulted in the removal of more articles, after which 69 articles remained. Finally, they read the entirety of the papers and removed 43 of them from their research, thus resulting in 26 papers. However, after reading the papers and their respective references, they decided that 3 additional papers should be considered into their research.

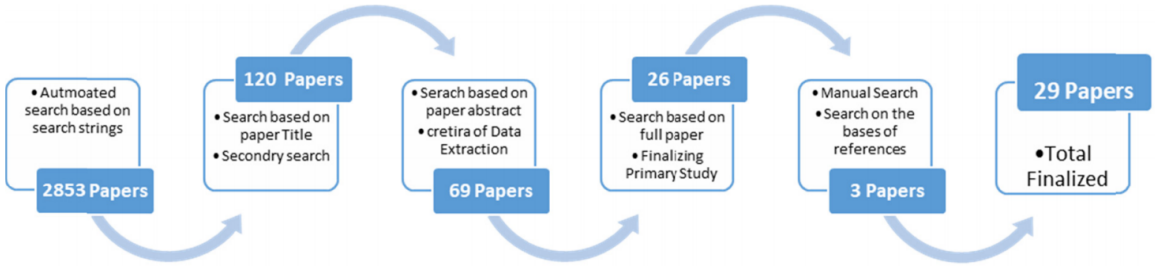


Figure 2: Research selection process

In the researched articles, various techniques of violence detection are proposed, and can be separated into 3 distinct categories: Violence Detection (VDT) using ML, VDT using SVM, and VDT using Deep Learning (DL).

3.2.1 *Violence Detection using Machine Learning*

In order to detect violence whilst using ML techniques, several methods have been presented, making use of a variety of different traditional ML algorithms such as K Nearest Neighbours (KNN), Adaboost, amongs others, as a classifier. Some of the methods that were researched are:

Motion Blob Acceleration Measure Vector (AMV) method for detection of fast fighting from video

In Gracia et al. (2015), researchers assume that in a fighting scene, the motion blobs have a specific shape and position. Having been analysed, the K largest are ultimately classified as either being violent or non violent.

so as to detect objects, this method used an Ellipse detection method. To extract features, an algorithm to find the acceleration was used, and finally spatio-temporal features were used in an effort to classify the scenes. This method was tested with both crowded and less crowded scenes, yielding a near 90% accuracy rate.

While this method was outperformed by State of the Art methods in terms of accuracy, its has expressively faster computation time, which does make it desirable for real-time applications.

Multiple anomalous activity detection

In Chaudhary et al. (2018), researchers developed three major stages so that anomalous activity could be detected.

Firstly, moving objects that are featured in the video stream are detected, whilst removing noise. The movement of said objects are then tracked, via a process of feature extraction, which is used so as to identify key features like direction, speed, dimensions and centroid. Lastly, the method of rule-based classification method is used to categorize the actions from videos.

In order to detect objects, this method used a Gaussian mixture model. To extract features, different formulas were applied on the consecutive frame so as to extract the required feature. As mentioned above, a rule-based classification was used to classify the scenes. This method was tested with less crowded scenes, scoring up to 90% accuracy rate.

According to the results of experiments, this method was able to detect various types of anomalous activities in different scenarios while attaining good accuracy.

SVM method recognition based on statistical theory without decoding of video frames

In Xie et al. (2016), researchers proposed a fast method of violent activity recognition that is based on motion vectors.

Firstly, motion vectors are directly extracted from the compressed video sequences, after which, the aforementioned vectors' attributes are analysed in each frame, and between the frames, in order to attain the Region Motion Vector (RMV) descriptor. Lastly, after the radial basis has been taken, using SVM as the kernel function to classify the RMV and decide whether or not violent activity is present or not.

To detect objects, a vector normalization method was used. To extract features, a macro block technique was employed, and finally so as to classify, the region motion descriptor was used. This method was tested in a crowded scenario and scored around 96% accuracy.

3.2.2 *Violence Detection using SVM*

Solve detecting problem by dividing the objective in depth and clear format using Con Net

In Peixoto et al. (2018), researchers used independent networks to learn features specifically related to violence, like blood, explosions, fights, etc. After that, in an effort to describe the violence while using such features, distinct SVM classifiers are trained for each of the concepts, and their results are later merged into a meta-classification.

To detect objects, movement detection and a Temporal-Robust Features (TRoF) model were used, and to extract features, a Bag of Words (BoW) method was used. This method was used in sparsely crowded scenarios, attaining a 96% accuracy rate.

Determine the occurence of violent purpose extended from of IVF and sliding windows

In Perronnin et al. (2010), researchers improved the Fisher Kernel framework, calling it Improved Fisher Vector (IFV), which resulted in an accuracy improvement of image classifiers. However, has it didn't have spatio-temporal analysis capabilities it wasn't suited to be used in video analysis. In Bilinski and Bremond (2016), researchers proposed an extension of IFV which gave it such capabilities, and boosted the IFV to achieve similar accuracy (whilst keeping the representation more compact), when compared to spatio-temporal grids.

3.2.3 *Violence Detection using Deep Learning*

DL is a subset of ML methods, based on ANNs. Its learning can be Supervised, Semi-Supervised or Unsupervised (Schmidhuber, 2015).

Among its most notorious architectures are: Convolutional Neural Networks (CNNs), Deep Neural Networks (DNNs), and Recurrent Neural Networks (RNNs). These have been applied to in a variety of fields, ranging from Computer Vision, Speech Recognition, and Board Game Programs, such as AlphaGo, a computer program which via extensive training from both computer and human play, was trained to play the board game *Go*, even beating the top player at the time (2016), Lee Sedol, securing a 4-1 win.

Detect Violent Videos using ConvLSTM

In Sudhakaran and Lanz (2017), a deep neural network based method is proposed to recognize violence in videos. A CNN is used to extract features from frame level in videos, said features are then accumulated using a variant of Long-Short Term Memory (LSTM) that uses convolutional gates.

Experiments performed on three popular datasets (Hockey, Movies, and Violent-Flows) reveal that the proposed model outperforms State of the Art methods like Violent Flows (ViF) + Oriented Violent Flows (OViF), ViF, three streams + LSTM in terms of accuracy.

This model was tested in a crowded scenario, where it got an accuracy rate of approximately 97%.

Hough Forest Methodology for reorganization

In Serrano et al. (2018), the researchers highlighted the necessity of violence detecting models being efficient. As such, they proposed a hybrid feature "handcrafted / learned" framework. Handcrafted spatio-temporal features are able to achieve high accuracy or both appearance and motion, however, the extraction of such features, is still troublesome for some applications.

The model first aims to get the illustrative image from the video sequence taken as input for feature extraction, and Hough Forest is used as a classifier. Afterwards, a 2D CNN is used to clasify the image. The propossed approach was tested in a less crowded scenario where it got accuracy rates ranging between 84% and 96%.

Violence Detection using Spatiotemporal Features with 3D CNN

In Ullah et al. (2019), a triple staged end-to-end DL framework that is capable of violence detection is proposed.

In the first stage, the researchers first detect people, making use of a CNN model to overcome and reduce the huge processing of unusable frames. They then feed a 3D CNN model 16 frames of input with detected individuals, where the spatio-temporal features are extracted and fed to the SoftMax classifier. Subsequently, the 3D CNN model is optimized using a neural networks optimization toolkit. After being tested on the Violent Crowd Itcher (2013), Hockey Nievas et al. (2011) and Violence in Movies Nievas et al. (2011) datasets, this approach was found to outperform State of the Art methods, and achieving an accuracy rate of approximately 97%.

4

TRAINING METHODOLOGY

In this chapter, the particularities surrounding the training methodology will be detailed, from the pipeline that will be used in order to train and assess how accurate the models are, to the data used to train these aforementioned models.

In these training sessions, each of the chosen pretrained models that will be used in each different stage of the pipeline will be tested, using different settings, in order to better comprehend which configuration better suits each model and yields the best results.

Each training session is composed of epochs, and each epoch of 2 stages, firstly the training stage and secondly the validation stage. In the training stage, each of the models will be fed data from which they'll try to extract knowledge, understanding what features of the fed data correlate to the label that defines it. In this case, what features lead to a video stream containing or not violence in it. In the validation stage, each of the models will be fed data, however, they will not learn from it, and instead try to leverage their previously gained knowledge in order to predict to what category does the data that its being fed belong to, in this case, whether the video stream contains or not violence. With each training session, data pertaining to each of the models in the pipeline and their performance will be saved in order to analyse how well the models behaved in their task of differentiating violent scenes from non violent ones. Such data will encompass accuracy percentage in each epoch, loss in each epoch, and also the necessary information to compose a Confusion Matrix (CM) (True Positives, True Negatives, False Positives and False Negatives) regarding the last epoch.

Since the main ambition of this Thesis is to test how well different modalities of ML perform in the detection of violence in a video stream, one must conduct the training in a slightly different way than it is usually conducted. Frequently, a training session in ML is comprised of loading a model, loading data, and then feeding that data to the aforementioned model, and recording its results in each epoch. Seeing that in this Thesis, 4 ways of detecting violence must be evaluated, the architecture of the training sessions will be slightly altered in order to best accommodate this change of paradigm.

4.1 TRAINING ARCHITECTURE

As was previously mentioned, the four different types of classifier that will be tested so as to see which performs better in violence detection are:

1. **Audio Classifier** - a model will analyze an Mel Frequency Cepstral Coefficient (MFCC) spectrogram and perform image classification on it;
2. **Video Classifier** - a model will receive a batch of images taken from a video stream;
3. **EF Classifier** - a EF model will receive both audio and video data, whose respective features will be extracted by an audio and a video model, and then said features will be merged and further analysed;
4. **LF Classifier** - a LF model will receive the outputs from audio and video models, proceeding to then analyse said outputs and determine if violence occurred or not;

Taking the training of these 4 models simultaneously into account, the architecture of the training program that was used was slightly changed, as was previously explained. One typically trains a model by loading the data and allowing the model to predict over it, however, since in this Thesis, the use of LF classifiers are being tested, it became necessary to save the predictions made by the audio and video classifiers in order to feed them to the LF classifier.

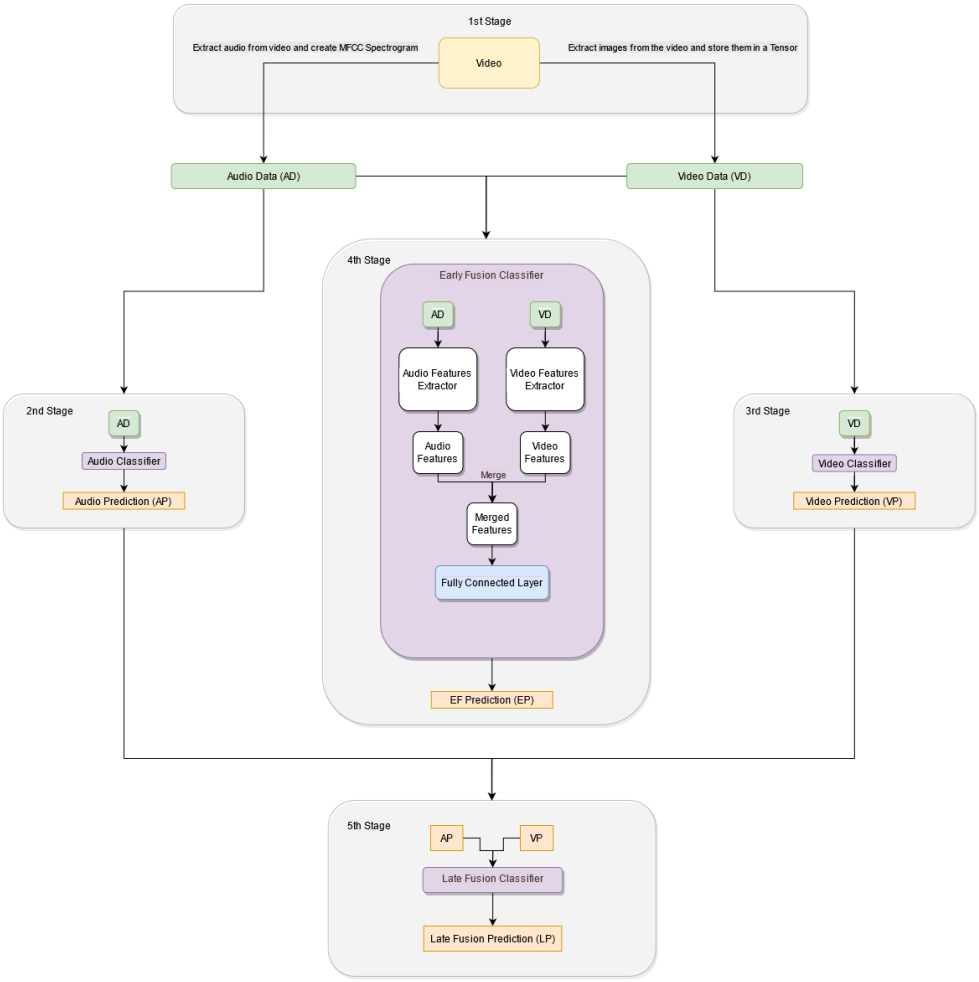


Figure 3: Training Architecture

In the first stage, the relevant data of each video is gathered, so the audio extraction and the extraction of frames from the video. In the second stage, the audio data is fed to the audio classifier which will receive an image of a MFCC spectrogram, over which it'll make its prediction. The third stage proceeds in the same fashion as the second, with the exception that it pertains to the video aspect of the analysis, and the video classifier will receive a Tensor with x frames of the video as input. The fourth stage is the most complex of them all, as both the audio and video data are fed to the EF classifier, which is itself composed of both an audio and video feature extractor. After two extractors process their respective data, the audio and video features are merged together, and passed onto a fully connected layer, which is responsible for the EF prediction. Lastly, in the fifth stage the predictions from both the audio and video classifier are fed to the LF classifier which will do its own prediction, after which the four different predictions produced by each of the different methods are available.

4.2 DATA

4.2.1 *Used Datasets*

In the Action Recognition portion of the ML field, these is no shortage of Action Recognition datasets, e.g. UCF50 Reddy and Shah (2013) and UCF101 Soomro et al. (2012), that feature respectively 50 and 101 categories of actions. As there are Datasets that feature non violent actions, there are also ones that features violent actions, some examples of these are the Hockey Fight Detection Dataset Abdali and Al-Tuma (2019), and Real Life Violence Situations Dataset Soliman et al. (2019), the last one containing a variety of violent scenes, ranging from punches, kicks, throws, among other violent actions.

For this Thesis, it was deemed that Hockey dataset would be unsuitable, as despite it showing violent actions, these scenes would perhaps not generalize well to violent acts committed by people in day to day situations due to the inherent characteristics to that dataset (voluptuous hockey equipment, bats, the ice ring in which they play). Hence, the decision was made to use the Real Life Violence Situations Dataset, however, after running into some errors, a problem became clear. Most of the videos featured in the dataset contained no sound whatsoever. It was then decided that the dataset should be exploited to its full potential, which resulted in around 250 videos, but nearly all of them violent, as very little of the non violent videos had sound.

To mend this, scenes from videos of the 'City Walking' genre were utilized. These videos, are filmed by people with GoPros or similar cameras, that do not speak, and limit themselves to just walking about different parts of different cities, and capturing footage and sound of everyday urban environments. In these videos, one can see people walking the streets, talking, eating, and engaging in mundane actions. Consequently, 'City Walking' videos from several cities were downloaded and excerpts from them made in order to give the models a wide range of non violent scenes with audio.

In total, 500 videos, with a duration of around 10 seconds each were collected, being that 250 featured violent scenes, and 250 non violent scenes. Besides the retrieval of public datasets featuring violent actions, the recording of videos specific to this project were also scheduled to take place, which feature the simulation of several violent scenarios. However, the recordings were involved in some bureaucracy which delayed their recording, so they're still being recorded, which rendered them unavailable to be used in this Thesis.

(a) (b)

Figure 4: Screenshots of videos in the Dataset

4.2.2 *Pre-Processing*

In ML, it's common practice that some kind of pre-processing will need to be done to the data that will be used to train the models, as some of its features may contribute negatively to the models' training.

One such example of these negative impacts regards images, as they are typically stored as multi-dimensional arrays of values ranging from 0 to 255. If one was to feed images as they are to a model, one possible consequence would be what is often referred to as 'Exploding Gradients', where a model's weights will vary dramatically from extremely high to extremely low values, possibly reaching NaN values, rending the model useless.

After normalizing the data, one typically ends up with data that either ranges from [0,1] but it's also possible that the values are normalized to a range of [-0.5,0.5]. In this case, the normalization was done in such a way that the final values ranged from [0,1].

4.2.3 *Data Augmentation*

Another important step in ML is that of Data Augmentation, which consists of using the data that one has available, and slightly altering as to make a dataset more robust. This is helpful, as it mitigates some issues, for example, unbalanced datasets.

A dataset will ideally be balanced in the sense that it'll have a certain number of classes, and each class will ideally have the same amount of data, for example, if the concerned dataset of videos concerning action recognition, and it contains 5 different actions to be recognized, then every action would need to have the same number of videos. This is important, because should the dataset be unbalanced, this can contribute to undesired tendencies in the model, and perhaps some bias. Data Augmentation can also be utilised in order to expand

datasets, since in some cases, one might have a smaller dataset than necessary, and in order to solve this, Data Augmentation can be used.

Some typical approaches when augmenting data are:

1. **Rotations** - the images are rotated by a certain amount of degrees, either left or right;
2. **Scaling** - the images are scaled either up or down;
3. **Mirroring** - the images are mirrored;

4.2.4 *Feeding the Dataset*

When feeding the dataset, some concerns must be addressed, one of the most common being the dataset size. More often than not, the dataset will be large, and due to hardware limitations, it will have to be split into smaller batches.

The Batch Size (BS) is a hyper-parameter that quantifies the size of the batch, in this Thesis case, how many videos will be fed at once to the model. The image batch will be propagated forward through the network in order to calculate the error between the network's predictions and the ground truth targets. Then, the network weights will be updated by back propagation, according to the error. Hence, the back propagation error is directly influenced by the number of samples of each batch, and also its heterogeneity Goodfellow et al. (2016).

High BS values can have troublesome consequences, for example, significant degradation in the quality of the model, as measured by its ability to generalize. The lack of generalization ability occurs due to the fact that large-batch methods tend to converge to sharp minimizers of the training function Keskar et al. (2016). On the other hand, a BS that is too small will lead to longer training times, hence, some tuning is needed.

4.3 TRAINING NEURAL NETWORKS

As was previously mentioned in Section 3.1.1, Neural Networks' training can be split into some different types, namely:

1. **Supervised Learning**: where the model is exposed to new labeled information, and attempts to learn what features are correlated to its label;
2. **Unsupervised Learning**: where the model is exposed to new unlabeled data and it looks for previously unnoticed commonalities in the data;
3. **Semi-Supervised Learning**: where the model is fed a combination of both labeled and unlabeled data;

In this Thesis, only Supervised Learning was taken into consideration, due to the availability of labeled data. There are also two possible approaches in order to train a Neural Network:

1. **Training from Scratch**: where the model is created with a random weight initialization;

2. **Transfer Learning**: where one takes a model that has previously been trained on another dataset, and has its weights adjusted to that aforementioned dataset. Many models are trained with State of the Art architectures on the ImageNet are available online Simonyan and Zisserman (2014). In order to adapt the model to one's use case, a change to the output layer is most likely necessary, and then a training session with the desired data;

5

MODELS

One of the most influential decisions regarding this Thesis, concerns which models are going to be utilised in each step of the pipeline of the project. As was previously explained in Section 4.1, the pipeline will consist of 4 types of classifiers, in order to determine which of these performs better.

For each classifier (except LF) several models were tested, in order to have more robust results from which to draw conclusions from. In this chapter, each of the models used for each of the classifiers will be explored, having their architecture detailed, its origins explained, so as to better understand them and why they are suitable for the task at hand.

Naturally, some criteria had to be set in order to determine which models are better suited to be used in each stage of the classifier. Said criteria, should reflect the needs of both the models, for example in the hardware requirements it depends upon, and the use case itself. In this Thesis, the selected criteria is:

i **Accuracy**: the most important factor to consider, is how accurate the model is, or how well it detects violence. This will be the ultimate objective, and most of the efforts detailed in this Thesis, will be directed to maximizing the models' accuracy;

ii **Hardware Requirements**: are also a relevant factor to consider, as if one considers a use case in which a model as to be implemented in a CCTV setting, it will most likely have some hardware limitations, in which case, a neural network that requires the most recent of GPUs is not adequate;

iii **Real Time Predictions**: the chosen model, should be able to analyze the video feed in real time, which in and of itself is a challenge, as it must do so swiftly, so any model that takes too long to predict, is automatically unsuitable for the use case at hand;

Attending to the aforementioned criteria, the following models were chosen, and will now be detailed.

5.1 AUDIO

Audio analysis using ML techniques based in extracting useful information from a sound track, analysing and then predicting over it. This useful information consists in the frequency and amplitude of the sound wave over a period of time. Two popular ways of extracting these features from an audio signal are:

1. Short Term Fourier Transformation (STFT): an audio waveform is converted to a spectrogram using STFT. This spectrogram displays the time-frequency changes as a 2D array of complex numbers that represent the magnitude and the phase;

2. MFCC: consists of calculating the power spectrum that gives the frequency spectrum to identify the present frequencies. To calculate the power spectrum, one must apply the Mel Scale filter bank, in order to extract the frequency bands. Afterwards, one must apply the log filter banks, and apply the Discrete Cosine Transformation (DCT) coefficients to generate a compressed version of filter banks, and achieving the MFCC;

In this Thesis, the MFCC approach is used.

The Audio classifiers being used in this Thesis were both proposed in He et al. (2016), along with a few other networks. The objective of the authors of this paper was to present a residual learning framework that would ease the training of networks that are substantially deeper than those that had been used thus far. Their architecture was created having, mainly, VGG nets in mind, having up to 8x the depth (when comparing the 152 layer version), and attempted to achieve similar if not better results, whilst having lower complexity.

The proposed neural networks were highly successful, having earned the authors the 1st place on the ILSVRC 2015 classification task, and having achieved better results than the VGG networks on the ImageNet classification dataset.

5.1.1 *Resnet18*

The first Audio Classifier being detailed is the Resnet18, with the following architecture:

Layer Name	Output Size	Resnet18
conv1	112×112	7×7, 64, stride 2
conv2_x	56×56	3×3 max pool, stride 2
		$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$
average pool	1×1	7×7, average pool
fully connected	1000	1000 fully connective
softmax	1000	

Table 1: Resnet18's Architecture

Looking at Table 1, it's possible to probe the architecture of the Resnet18 model. This model's architecture starts with a convolutional layer with 7x7 kernel size, and is followed by the beginning of the skip connection. The input from there is added to the output that is achieved by the 3x3 max pool layer, and 2 pairs of convolutional layers with a kernel size of 3x3, having each 64 kernels. This part represents the first residual block, and 5 convolutional layers in total.

From there, the output of this residual block is passed on to 3 additional residual blocks, each with 2 pairs convolutional layers. The convolutional layers of these residual blocks have a kernel size of 3 x 3 each, and 128 such filters. The convolutional layers in each residual block see their number of filters double, comparatively with the convolutional layers in the previous block, and the size of its output decrease by half. With these additional convolutional layers and the fully connected layer that is featured afterwards, the total amount of layers is brought up to 18, from where the model get its name.

5.1.2 *Resnet34*

Layer Name	Output Size	Resnet34
conv1	112×112	7×7, 64, stride 2
conv2_x	56×56	3×3 max pool, stride 2
		$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$
average pool	1×1	7×7, average pool
fully connected	1000	1000 fully connective
softmax	1000	

Table 2: Resnet34's Architecture

The second network that will serve as Audio Classifier is the Resnet34. Its design, follows that of the remaining proposed neural networks. As we can see in Table 2, its architecture closely resembles that of the Resnet18. This model's architecture begins by having almost the same setup for its first convolutional layer, and residual block following it, with the difference that this residual block features 3 pairs of convolutional layers, and not 2. Afterwards, it also follows the same trend, with the convolutional layers of each residual block seeing their number of filters double, when compared with the convolutional layers in the previous block, but in this model's case, each residual block doesn't necessarily have only 2 pairs of convolutional layers. More precisely, the second residual block features 4 pairs of convolutional layers, so 8 in total. From there, the trend is kept, but with the difference that the third residual block features 6 pairs of convolutional layers, and the fourth, 4 pairs.

5.2 VIDEO

Video analysis presents different challenges to Image analysis. When analysing a video, one must take into account the previous events that occurred in the sequence, so as to take into account the whole video, and not individual frames.

In order to do this, one must not only take into account the 2 dimensions that are present in Image analysis, but also a third dimension, the temporal dimension. To accommodate this new dimension, several strategies are available, however, the one being employed in this Thesis revolves around 3D CNN, which as shown in Tran et al. (2018), outperform 2D CNN, on challenging action recognition benchmarks, such as Sports-1M (Karpathy et al. (2014)) and Kinetics (Kay et al. (2017)).

The authors of Tran et al. (2018) were inspired by these promising results, and introduced 2 new forms of spatiotemporal convolutions that can be viewed as middle grounds, between the extremes of 2D (spatial convolution) and full 3D (spatiotemporal convolution). Their first proposal was called Mixed Convolution (MC), and is explained in Subsection 5.2.1.

5.2.1 *Resnet_MC18*

This proposal consists of employing 3D convolutions solely in the early layers of the network, with 2D convolutions in the top layers, the rationale being that the motion modelling is a low/mid-level operation that can be implemented via 3D convolutions in the early layers of a network, and spatial reasoning over these mid-level motion features (implemented by 2D convolutions in the top layers) lead to accurate action recognition, and in this Thesis, to the accurate recognition of violent actions Tran et al. (2018).

Layer Name	**Output Size**	**Resnet MC18**
conv1	$16 \times 56 \times 56$	$3 \times 7, 64$, stride 2
conv2_x	$16 \times 56 \times 56$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$
conv3_x	$16 \times 28 \times 28$	$\begin{bmatrix} 1 \times 3, 128 \\ 1 \times 3, 128 \end{bmatrix} \times 2$
conv3_x	$16 \times 14 \times 14$	$\begin{bmatrix} 1 \times 3, 256 \\ 1 \times 3, 256 \end{bmatrix} \times 2$
conv3_x	$16 \times 7 \times 7$	$\begin{bmatrix} 1 \times 3, 512 \\ 1 \times 3, 512 \end{bmatrix} \times 2$
average pool	1 ×1 ×1	7 ×7, average pool
fully connected	400	400 fully connective
softmax	400	

Table 3: Resnet MC18's Architecture

As it's possible to verify by going through the Table 3, this model's architecture begins with a convolutional layer with 3x7 kernel size, and is followed by the beginning of the first residual block. Similar to the previous

models, this block features pairs of convolutional layers, as do the remaining 3 blocks, with each of the blocks having 2 pairs of convolutional layers each. These residual blocks also follow the trend of the previous models, in the sense that with each block, the amount of filters each layer features, doubles. This models features an average pool after the 4 residual blocks, which is then followed by a fully connected layer.

5.2.2 *Resnet(2+1)D*

The second proposal by the authors of Tran et al. (2018) is a "(2+1)D" convolutional block, which explicitly factorizes 3D convolution into 2 separate and successive operations, a 2D spatial convolution and a 1D temporal convolution, justified by the authors with 2 reasons.

The first advantage is that having an additional nonlinear rectification between these 2 operations, effectively doubles the number of nonlinearities compared to a network using full 3D convolutions for the same number of parameters, thus rendering the model capable of representing more complex functions. The second potential benefit, is that the decomposition facilitates the optimization, yielding in practice both a lower training loss, and a lower testing loss, hence effectively making the (2+1)D blocks easier to optimize.

Layer Name	**Output Size**	**Resnet (2+1)D**
conv1	$16 \times 56 \times 56$	1×7, 45, stride 2
conv2	$16 \times 56 \times 56$	3×1, 64, stride 2
conv3_x	$16 \times 56 \times 56$	$\begin{bmatrix} 1 \times 3, 144 \\ 3 \times 1, 64 \\ 1 \times 3, 144 \\ 3 \times 1, 64 \end{bmatrix} \times 2$
conv4_x	$16 \times 16 \times 28$	$\begin{bmatrix} 1 \times 3, 230 \\ 3 \times 1, 128 \\ 1 \times 3, 230 \\ 3 \times 1, 128 \end{bmatrix} \times 1$
conv5_x	$8 \times 8 \times 28$	$\begin{bmatrix} 1 \times 3, 288 \\ 3 \times 1, 128 \\ 1 \times 3, 288 \\ 3 \times 1, 128 \end{bmatrix} \times 1$
conv6_x	$4 \times 14 \times 14$	$\begin{bmatrix} 1 \times 3, 460 \\ 3 \times 1, 256 \\ 1 \times 3, 460 \\ 3 \times 1, 256 \end{bmatrix} \times 1$
conv7_x	$4 \times 14 \times 14$	$\begin{bmatrix} 1 \times 3, 576 \\ 3 \times 1, 256 \\ 1 \times 3, 576 \\ 3 \times 1, 256 \end{bmatrix} \times 1$

conv8_x	$4 \times 7 \times 7$	$\begin{bmatrix} 1 \times 3, 921 \\ 3 \times 1, 512 \\ 1 \times 3, 921 \\ 3 \times 1, 512 \end{bmatrix}$ ×1
conv9_x	$2 \times 7 \times 7$	$\begin{bmatrix} 1 \times 3, 1152 \\ 3 \times 1, 512 \\ 1 \times 3, 1152 \\ 3 \times 1, 512 \end{bmatrix}$ ×1
average pool	1 ×1 ×1	7 ×7, average pool
fully connected	400	400 fully connective
softmax	400	

Table 4: Resnet (2+1)D's Architecture

Looking at the Table 4, one is able to verify that this model's architecture begins with a convolutional layer with 1x7 kernel size, which is followed by a second convolutional layer, with a 3x1 kernel size. After these 2 layers, the first residual block is found. Unlike previous models, this block features 2 groups of 4 convolutional layers, while the remaining residual blocks feature only one group of 4 convolutional layers. These residual blocks also follow the trend of the previous models, in the sense that with each block, the amount of filters each layer features, increases, but in these case, they don't double with each residual block. This model also features an average pool after its residual blocks, which is then followed by a fully connected layer.

5.3 EARLY FUSION

The EF concept has been used for sometime, being proposed in such scenarios as described in Snoek et al. (2005). As was previously explained in Subsection 3.1.4, EF consists in extracting both the audio and video features, and merging them in a certain way, and then analysing the combined data.

In this Thesis, the audio features are extracted and presented as the image of an MFCC spectrogram, and the video features are presented as a multi-dimensional array with the shape (3 x T x 224 x 224), with 3 representing the RGB colors, T being the number of frames in a clip, and 224 being both the height and width of the frames. After being collected, they are fed to the model, which in turn makes its prediction over the data.

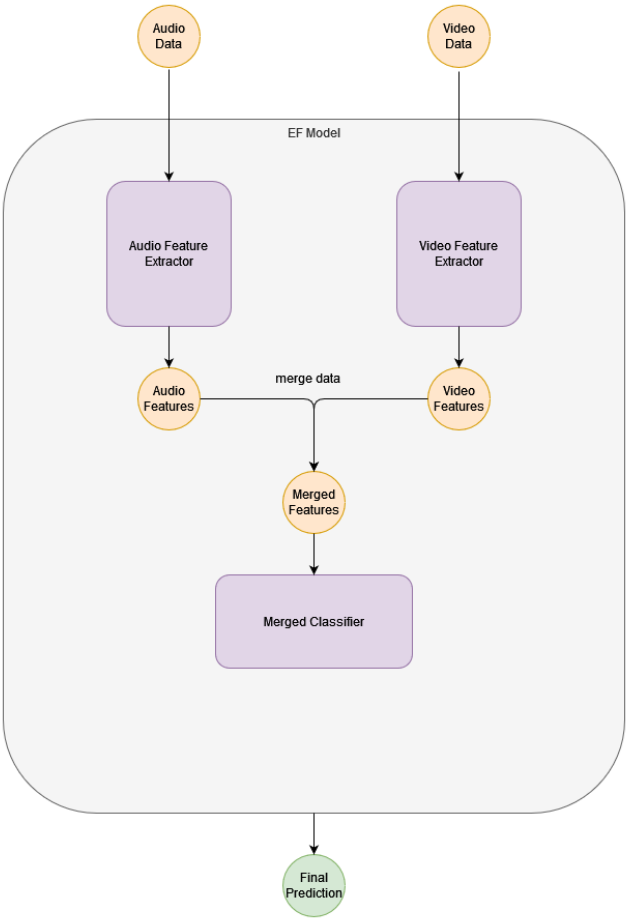


Figure 5: Early Fusion Diagram

Looking at Figure 5, one can get a clearer picture of the model's architecture. Firstly, the Audio and Video data are loaded in order to be fed to EF model. This model consists of 2 feature extractors, one for Audio, and the other for Video. In this Thesis' training period, every time that new models were tested, the feature extractor part of their architecture will be used in the EF Model. These will be fed their respective data, which they'll process, and extract their relevant features. After both components have handled their data, the resulting Audio and Video extracted features are merged. This merging can be done in several ways, however, in this Thesis they are simply averaged out. After being merged, they are fed to a classifier which will do the final prediction.

5.4 LATE FUSION

Like EF, LF has also been present in the ML field for quite some time. Of course, contrarily to EF, in LF, one allows the audio and video classifiers to be fed their respective data, and make their prediction over it, and afterwards, merging the resulting predictions.

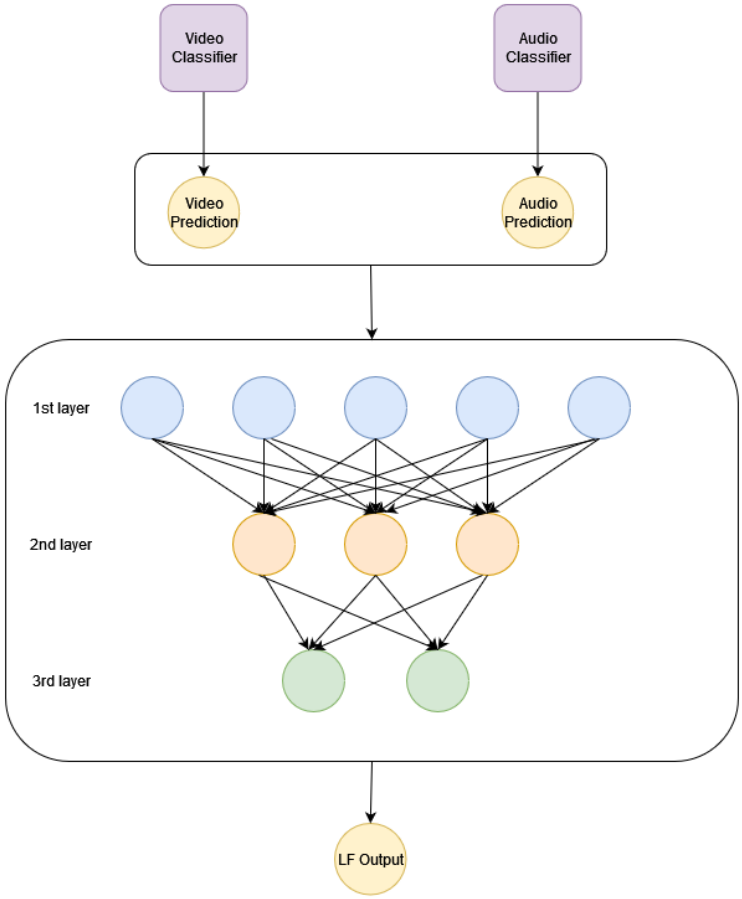


Figure 6: Late Fusion Diagram

This can be better understood by looking at Figure 6. It's possible to see the Audio and Video classifiers in the top of the image. These are the aforementioned Resnet18 / Resnet34 and Resnet_MC18 / Resnet(2+1)D models, respectively. These classifiers will both receive their respective data and predict on it. These predictions, will be then merged by averaging them out, and will be fed to the LF model. This model consists of 3 linear layers, so it has a simple architecture. It can afford to be so simple, due to the fact that the heavier part of the processing is done by the Audio and Video classifiers.

6

RESULTS AND DISCUSSION

In this Chapter, two of the most important sections of the Thesis are found, namely 6.1 and 6.2. Firstly, in Section 6.1, all of the results will be compiled, and analysed in order to allow the user to draw comparisons, and understand in which way each different hyper-parameter and configuration effect the model's performance. In 6.2, the best results by each classifier type are displayed, in an effort to more easily exhibit their best performances, in which conditions they occurred.

6.1 RESULTS

After having completed the training of every classifier, in diverse conditions, altering the different hyper-parameters, a set of results was achieved. These results encompass the close to 200 hours of training, divided by 8 training sessions, in which the 4 different models were used in their respective classifiers. These results are comprised by dozens of graphs and text files that detail, ranging from CMs that plot the last epoch of the training session, to line graphs that plot the evolution of the accuracy and loss rates. Having this data available is crucial in order to determine what approach is working best, and if something's not performing to its fullest extent, why that is.

From these results, it was possible to assemble the Table 5:

Classifier Type	**Average Training Accuracy**	**Average Train Loss**	**Average Validation Accuracy**	**Average Validation Loss**
Audio	86.75	0.21	70.5	0.81
Video	90.5	0.08	86	0.25
Early Fusion	90.38	0.09	84.25	0.3
Late Fusion	81.25	0.54	74.75	0.57

Table 5: General Classifier Results

Somewhat unsurprisingly, the Video classifiers were the ones who performed better in all metrics, having attained on average 86% of accuracy in the validation set. As well as having the highest accuracy rate in the Validation set, these also scored the lowest loss in the Validation set, performing according to expectations,

as was expected that the Video classifiers performed well. One surprise, however, was that the EF classifiers performed better than previously expected, having achieved on average 84.25% of accuracy in the Validation set, a result that differs very slightly from the Video classifiers' score, while having recorded also a small loss in the Validation set. Furthermore, the Audio and LF classifiers performed as expected, not having achieved great accuracy rates on the validation set, but revealing themselves quite capable on the training set. These results fall in line with the expectation that the Audio and LF classifiers would have worse performance than the Video classifier, but it was initially thought that they'd perform more or less along the lines of the EF classifiers, however, as was previously said, these classifiers outperformed the expectations set for them.

In the following sub-chapters, the results achieved by the 4 types of classifiers tested in this Thesis will detailed and analyzed. They'll be evaluated under several perspectives, and connections between the different settings related to the hyper-parameters and the performance of the classifiers will be attempted to be drawn. Firstly, the average values of each of the models used for the classifier in question will be analysed, regardless of the hyper-parameters set. Afterwards, the hyper-parameters will be taken into account, in order to get a more comprehensive look at the results, in a better effort to understand if any conclusions can be drawn.

6.1.1 *Audio*

The theory behind the classifying of Audio arises from the sound characteristics that will typically be present in violent acts, such as screaming, battering of surfaces, breaking of items, and so forth. It's believed that the classifier will be able to identify in which scenarios violent acts are occurring simply based on the audio feed.

In the Table 6, a general look into each of the models' performance can be had.

Model Name	**Average Training Accuracy**	**Average Train Loss**	**Average Validation Accuracy**	**Average Validation Loss**
Resnet18	86.75	0.21	70.5	0.81
Resnet34	85.5	0.21	66.75	0.84

Table 6: General Audio Classifier Results

From there we can see that in a general sense, the models are extremely similar, which is to be expected, as they were conceived by the same people, and are based on the same architecture, the only difference being is that the Resnet34 has more layers than the Resnet18 one. All of their values are very similar, with perhaps the Average Validation Accuracy being the one that has the most significant difference, but even in that case it's a 3.75% difference, which is very similar.

When comparing the results from Table 6 with those from Table 7, it's possible to realise a trend, in which both models performed better with a BS of 7 than 5, mainly in average accuracy. This is particularly notable with the Resnet34, which scored 71% of average accuracy in the validation set, almost 5% more than its average validation accuracy, as seen in Table 6.

Model Name	Batch Size	Average Training Accuracy	Average Train Loss	Average Validation Accuracy	Average Validation Loss
Resnet18	5	85.5	0.25	70.5	0.69
Resnet18	7	88	0.17	70.5	0.94
Resnet34	5	82.5	0.27	65.5	0.84
Resnet34	7	88.5	0.14	71	0.85

Table 7: Audio Classifier Results by Batch Size

When looking at Table 7, it's possible to verify that a wider range of results are present. When comparing the models' performance with the different BSs, the results are somewhat inconclusive. In one hand, the average training accuracy somewhat improves, being accompanied by a drop in the average training loss. However, when it comes to the average accuracy in the validation set, in the Resnet18 model's case there was no improvement gained from increasing the BS from 5 to 7, having even had a worse performance, as the average loss in the validation set increased. In the Resnet34's case, the average accuracy in the validation set improved by almost 6%, however the average loss barely increased.

Taking a look at the models' results from a Learning Rate (LR) perspective, it's possible to consult the Table 8

Model Name	Learning Rate	Average Training Accuracy	Average Train Loss	Average Validation Accuracy	Average Validation Loss
Resnet18	1e-3	90	0.11	73.5	0.59
Resnet18	1e-2	83.5	0.31	67.5	1.04
Resnet34	1e-3	90	0.09	69	0.73
Resnet34	1e-2	81	0.32	67.5	0.96

Table 8: Audio Classifier Results by Learning Rate

In this table, it's possible to verify how the LR effected the models' performance. When looking at the performance over the training set, both models performed similarly, scoring close to 90% accuracy with a LR of 1e-3, and having their accuracy drop to around 80% with a LR of 1e-2, and with this drop, an increase of the average loss ensued. Furthermore, the models trend in the training set stayed the same in the validation set as well. With an increase in LR, the models' performance decreased, verified in both the average accuracy with a decrease and an increase in the average loss.

It's possible, once again, to notice a trend when comparing the results from the Tables 6 and 8, that shows us that both the models benefit from using a LR of 1e-3, which it's possible to tell by both the increase in average accuracy in the training and validation set, and the accompanying loss reduction.

In a general sense, the Audio classifiers performed as expected, displaying a solid performance in general, but not performing as well as their other counterparts. The best result was achieved by **Resnet18** model, with

a LR of 1e-3 and a BS of 7, with it having achieved an average accuracy of 91% and an average loss of 0.08 on the training set, and seeing those values worsen to 76% and 0.51 in the validation set. Despite having fallen 15%, the average accuracy in the validation set still falls within the expected values. All of these results can be checked in the Figures 8 through 14.

6.1.2 *Video*

Moving on to Video classifiers, the expectations shift, as essentially all models used in State of the Art papers are Video classifiers, so these kind of models are expected to perform on a extremely high level. It's natural that Video classifiers are the kind that perform better, as the video features are quite more descriptive and correlatable to violent actions like punching, beating, of similar actions than the sound that these actions produce.

Looking at Table 9, the distinction between the performance of the Audio and Video classifiers becomes clear. In the training set, the Video classifiers outscored the Audio classifiers on average by around 5%. Their performance becomes even more impressive when it comes to the average accuracy on the validation set, in which the Video classifiers outscore their Audio counterparts by a whooping 15%. This is of course, to be expected, but still needs testing and proving.

Model Name	**Average Training Accuracy**	**Average Train Loss**	**Average Validation Accuracy**	**Average Validation Loss**
Resnet(2+1)D	90.5	0.08	86.25	0.24
Resnet_MC18	90.5	0.085	85.75	0.25

Table 9: General Video Classifier Results

The general results from the Video classifiers showed promise, so it's interesting to understand how the different parameters interact with each of the models that were chosen to be Video classifiers and their respective performance. Let's, firstly, take a look at their behaviour, from the perspective of the BS parameter.

Model Name	**Batch Size**	**Average Training Accuracy**	**Average Train Loss**	**Average Validation Accuracy**	**Average Validation Loss**
Resnet(2+1)D	5	90	0.09	86.5	0.23
Resnet(2+1)D	7	91	0.07	86	0.26
Resnet_MC18	5	90	0.1	85	0.25
Resnet_MC18	7	91	0.07	86.5	0.24

Table 10: Video Classifier Results by Batch Size

These results were quite surprising, due to their unexpected consistency, which seems to indicate that the BS of 5 and 7 are adequate values with which to train the models with. Across both models, regardless of which BS was used, the results remained remarkably close, with a maximum difference between the models' average accuracy in the validation set of just 1.5%.

Moving forward to the Table 11, it's possible to verify the LR's influence in the models' results.

Model Name	Learning Rate	Average Training Accuracy	Average Train Loss	Average Validation Accuracy	Average Validation Loss
Resnet(2+1)D	1e-3	90	0.08	89	0.13
Resnet(2+1)D	1e-2	91	0.08	83.5	0.36
Resnet_MC18	1e-3	90.5	0.09	89	0.12
Resnet_MC18	1e-2	90.5	0.08	82.5	0.37

Table 11: Video Classifier Results by Learning Rate

In this table, much more diverse results are present, which is quite apparent, but not right away. In regards to the average accuracy in the training set, neither the BS, nor the LR had too much of an effect, with every model scoring between 90% and 91% and an accompanying low loss. When it comes to the average accuracy in the validation set, however, the scenario changes more dramatically. Despite not having an extreme fluctuation, a close to 7% difference in the average accuracy between the models now exists, with it ranging from 82.5% to 89%. In this regard, it's clear to see that the LR bears a lot more relevance to the models' performance than the BS, at least in the used values.

The best results, regarding the average accuracy in the validation set, were achieved by both the Resnet(2+1)D and Resnet_MC18, having both of the models attained scores of 89% accuracy in the set. This impressive accuracy was obtained, as can be expected from the above explanation, with a LR of 1e-3. Comparing these results with the audio classifiers, it becomes clear why Video classifiers are used in virtually every State of the Art paper. Even when taking into account all the different testing scenarios, in no circumstance did the Audio classifiers score close to their Video counterparts. All of these results can be checked in the Figures 15 through 22.

6.1.3 *Early Fusion*

Moving along to the EF classifiers, it's easy to have little expectations for such a type of classifier. These are not at all widely used, not because they have little adaptability, because these can be used in a wide variety of scenarios, but perhaps because of the resources that these consume. In this Thesis, all but the last layers of the Audio and Video classifiers were used as feature extractors in the EF classifiers, so in practice, one would need to have double the processing capability in order to accommodate such a resource intensive model. On a small note, regarding the naming scheme for the EF classifiers, the name featured on the table follows the naming

scheme of Audio Model, followed by an underscore (_), followed by the name of the Video Model that were used as feature extractors.

After a quick analysis of the Table 12, one can notice some similarities with Table 9, but with slightly lower average accuracy in the validation set, however, these are still excellent results that surpassed any previously held expectations.

Model Name	Average Training Accuracy	Average Train Loss	Average Validation Accuracy	Average Validation Loss
Resnet18_Resnet(2+1)D	90.25	0.1	84.25	0.3
Resnet34_Resnet_MC18	90.5	0.09	84.25	0.29

Table 12: General Early Fusion Classifier Results

When compared to the General Audio Classifiers' results from the Table 6, these become even more impressive, since the EF classifiers make use of both Audio and Video models as feature extractors, and they seem to make a good use of both these classifiers, as despite the average accuracy in the validation set having gone down, it didn't do so by a huge amount, but by around 4%. This is quite interesting, because it appears to indicate that the classifier gave more influence to the input originating from the Video classifier, because if that was not the case, the average accuracy in the validation set would be around the 80% mark.

Looking at Table 13, one's able to verify how the BS interacts with the models' results. Interestingly, it interacts differently with each models, as it's clear to see when looking at the average accuracy over the validation set.

Model Name	Batch Size	Average Training Accuracy	Average Train Loss	Average Validation Accuracy	Average Validation Loss
Resnet18_Resnet(2+1)D	5	89.5	0.12	87	0.31
Resnet18_Resnet(2+1)D	7	91	0.08	81.5	0.29
Resnet34_Resnet_MC18	5	90	0.11	84	0.29
Resnet34_Resnet_MC18	7	91	0.08	84.5	0.29

Table 13: Early Fusion Classifier Results by Batch Size

In the classifier that uses the Resnet18 as audio feature extractor and the Resnet(2+1)D as video feature extractor, having a BS of 5 greatly benefits its average accuracy over the validation set, while having a BS of 7 hinders it slightly, having a difference of 6% of average accuracy on the validation set between the classifier with different configurations. On the other hand, with the classifier using Resnet34 and Resnet_MC18 as feature extractors, it barely affects it, having a mere difference of 0.5% average accuracy on the validation set.

Moving along to Table 14, a similar scenario happens. In this table, similarly to the previous one, when the hyper-parameter increases, the classifier that uses Resnet18 and Resnet(2+1)D as feature extractors sees its average accuracy over the validation set decrease as the LR increases from 1e-3 to 1e-2. When the classifier

uses Resnet34 and Resnet_MC18, on the other hand, it also affects the average accuracy on the validation set, but in a slighter way, with the accuracy only differing by about 2.5%.

Model Name	Learning Rate	Average Training Accuracy	Average Train Loss	Average Validation Accuracy	Average Validation Loss
Resnet18_Resnet(2+1)D	1e-3	90	0.09	87	0.175
Resnet18_Resnet(2+1)D	1e-2	90.5	0.1	81.5	0.425
Resnet34_Resnet_MC18	1e-3	90.5	0.085	85	0.22
Resnet34_Resnet_MC18	1e-2	90.5	0.1	83.5	0.36

Table 14: Early Fusion Classifier Results by Learning Rate

The best result, was achieved by the the classifier using Resnet18 and Resnet(2+1)D, with a BS of 7 and a LR of 1e-3, that attained an average accuracy of 88% on the validation set. This result was better than was expected of this classifier. It's even better when taking into account that the best result achieved so far by all of the tested classifiers was of 89% average accuracy on the validation set, a mere 1% difference, by both of the video classifiers. This is quite interesting, as despite the classifier making use of both Audio and Video classifiers, and the Audio classifiers having had much lower average accuracy that their Video counterparts, the EF classifier somehow managed to not be too hindered by using Audio classifiers. All of these results can be checked in the Figures 23 through 30.

6.1.4 *Late Fusion*

Lastly, the LF classifier will be analysed. This classifier differs from the other, because it was the only one which only had one type of model being tested, previously explained in Section 5.4 when all the other types of classifiers had two. This is both good and bad, as it'll provide better certainty of the results, as double the amount of tests will be done for a certain configuration of hyper-parameters, however, it's bad in a sense because only one model will be tested, instead of having more diverse models being tested.

Generally, the classifier didn't do neither too bad, nor too good. It only scored on average 81.25% of accuracy over the training set, quite lower than any of the other classifiers. When looking, however, at the average accuracy over the validation set, the values show more promise, with the classifier having scored on average 74.75% of accuracy over the validation set.

Average Training Accuracy	Average Train Loss	Average Validation Accuracy	Average Validation Loss
81.25	0.54	74.75	0.57

Table 15: General Late Fusion Classifier Results

These results are, naturally, far from the best that were recorded during this Thesis' testing, but as expectations were low for this classifier, these results are about what was expected of the classifier.

Looking at Table 16, it's possible to see how the different BS affect the classifiers' training and results. In this particular case, it's possible to see that the classifier benefits from having a BS of 5, as it sees its average acuraccy over the validation set increase by close to 4% to 79%, while having a BS of 7 hinders its performance, seeing it drop by close to the same amount, 4%.

Batch Size	Average Training Accuracy	Average Train Loss	Average Validation Accuracy	Average Validation Loss
5	88.5	0.53	79	0.56
7	74	0.55	70.5	0.58

Table 16: Late Fusion Classifier Results by Batch Size

Training the classifier with a BS of 5 greatly benefits its performance, resulting on an average accuracy over the validation set a bit lower than those achieved the Video and EF classifiers in certain training scenarios.

Moving on to the Table 17, one's able to see that similarly to the previous table, the changing of the hyper-parameter lead to a difference of behaviour by the classifier, but in this case, it impacted the models in the opposite fashion, with the model benefiting from an increase in the hyper-parameter.

Learning Rate	Average Training Accuracy	Average Train Loss	Average Validation Accuracy	Average Validation Loss
1e-3	77.5	0.57	71.75	0.6
1e-2	85	0.51	77.75	0.54

Table 17: Late Fusion Classifier Results by Learning Rate

The best result achieved by the classifier was attained with a BS of 7 and LR of 1e-2, combining the best of both configurations of the hyper-parameters as previously seen for this classifier. This setup resulted in 92% of average accuracy on the training set, which went down by 7% to 85% in the validation set. These results are better than expected because, as it was previously explained, expectations for this classifier were not high, so scoring a few percentage points below high achieving classifiers such as Video is an excellent result. All of these results can be checked in the Figures 31 through 38.

6.2 DISCUSSION

After all the tests are conducted and analysed, one is finally able to conclude which model and type of classifier performed best, by looking at Table 18.

Classifier Type	Model Name	Batch Size	Learning Rate	Average Validation Accuracy
Audio	**Resnet18**	7	1e-3	76
Video	**Resnet(2+1)D**	-	1e-3	89
Video	**Resnet_MC18**	-	1e-3	89
EF	**Resnet18_Resnet(2+1)D**	7	1e-3	88
LF	**-**	7	1e-2	85

Table 18: Best Average Accuracy on the Validation Set by Classifier Type

In this table it's possible to see what model, by type of classifier did scored the highest average accuracy rates over the validation set. A few caveats regarding this table are the fact that, as previously explained, only one model was tested as LF classifier, hence the '-' in its model name. The second caveat, concerns the '-' in the BS column of the Video Classifiers, which is due to the fact that with either a BS of 5 or 7, both of these models scored the same average accuracy of 89% over the validation set, so, the '-' signifies that regardless of the BS, the models performed the same, when having a LR of 1e-3.

Looking at the table, it's clear that all the models outperformed the expectations that were previously had for them. The Audio classifier, despite being the lowest scorer of the tested classifiers, still achieved a respectable 76% of average accuracy, which in and of itself is a good achievement. When we look however, at the other classifiers, one is able to truly perceive the potential that these models have for Action Recognition, and Violence Detection in particular. All of these models scored on average, between 85% and 89% accuracy on the validation set, which are remarkable results, taking into account both the time constraints that concerned the training, since these were trained in a shared machine which had to a lot of times be forfeited so that others could conduct training relevant to their research, and the constraints related to not having a comprehensive enough of a dataset. If this had not been the case, better results could have possibly been achieved, but these are still noteworthy results when taking into account all of the circumstances.

7

CONCLUSIONS AND FUTURE WORK

7.1 CONCLUSIONS

When concluding a Thesis, one must look back at the previously defined Research Questions, in order to answer that in a satisfactory manner. These questions will now be revisited, one at a time, and given an answer.

Research Question nr. 1: What methodologies exist and are now being leveraged in an effort to detect violence in video streams?

Research Answer nr. 1: Several methodologies are being used, at the present time, in order to recognize actions, and also to detect violence. Such methods include, but are not limited to the ones tested in this Thesis, although, the most popular and successful ones correspond to the ones featured in this research. Some are not so prominent, or successful, as is the case for example with EF Classifiers, but these still have some use.

When it comes to the objectives, these were also achieved:

— *Analysis of the existing methodologies and subsequent choice of the one that best suits the problem at hand* - Several methods were analysed, and according to the results acquired from all the conducted tests, one reaches the conclusion that the type of classifier that is better suited to detect violence is the Video Classifier, as was previously expected.

— *Requirements Elicitation, so as to know what limitations this work presents* - most classifiers were limited in some extent, whether it be by the achieved results, as was the case with Audio Classifiers, which performed better than expectations, but did not supersede other more accurate classifiers. This in and of itself, can be considered a limitation, as can the resources that the model requires. In this regard, the classifier that is more affected by this limitation is the EF Classifier, as it requires essentially having 2 models stored in memory, as was previously explained in Subsection 6.1.3.

— *Retrieval of scientific articles that feature State of the Art methods, in an effort to learn how the problems identified can be overcome* - Several scientific articles were analysed, as to realise how these problems that were identified can be overcome. Some are simply not possible to overcome, as for example the amount of resources that some models need to be executed, however, low accuracy over the validation set has some fixes, for example a more comprehensive training set, or a different hyper-

parameter setting, as it was seen that these can have severe effects in the final outcome of the model's training.

Research Question nr. 2: Which data should be used to train the model, and what characteristics should it possess?

Research Answer nr. 2: The data that's better suited to train the model should possess some features, among which, be comprehensive in the scenarios it covers, featuring a wide array of types of actions, in different scenarios, in order to train the model in the most diverse amount of settings possible.

To answer this question, the following objectives related to the collection and setup of data in order to train the model were highlighted, and are now answered:

— *Requirements Elicitation, in order to understand what characteristics the dataset should possess so as to better train the model* - as was described in several of the read scientific articles during this Thesis, and as was described immediately before, the dataset should contain some desirable features, such as having a comprehensive set of actions, among others, in order to better prepare the model for a diverse set of actions.

— *Explore existing datasets, disposing of those that do not bear the aforementioned attributes* - Several datasets that are used in action recognition scientific articles were found, for example the Hockey Dataset Nievas et al. (2011), the Violent Crowd Itcher (2013) or the Violence in Movies Nievas et al. (2011), but for previously explained reasons none of these were suitable to be used in this Thesis, leading to the decision of creating a dataset that suited this Thesis' needs.

— *Setup the chosen dataset, so that it can be used to train and test the model* - This objective was also achieved, as several preparations were made in order to setup the dataset, as was explained in Chapter 4.

Research Question nr. 3: How should the model be developed so that it attains the expected accuracy rates?

Research Answer nr. 3: From the research that was conducted in this Thesis, the configuration that achieved the best accuracy concerned the Video Classifiers, with either of the models that were tested, and having them be trained with a LR of 1e-3 and the BS bearing no relevance, as with either a BS of 5 or 7, 89% of accuracy over the validation set was achieved.

To answer this question, the following objectives related to the development of a ML model that is able do detect violence in a video stream were highlighted, and will now be answered:

— *Requirements Elicitation, so as to know what objectives the model should accomplish* - from a point of view of expectations, all the models achieved or superseded the expectations laid out for them, as could be seen in the Table 18, the table with the lowest performance scored on average 76% on the validation set, and that can be considered an outlier. When one verifies all the types of classifiers

except the Audio, all the average accuracies were situated between the 85% and 89%, which is quite remarkable in and of itself.

— *Analysis of current technologies, in order to decide which best suit the established development needs* - as was explained, the best results achieved overall were achieved by the models used in the Video classifiers, which both scored an average accuracy of 89% on the validation set, so it only makes sense that one of these models is the best suited to detect violence.

7.2 FUTURE WORK

ML as whole, came to revolutionize a diversity of fields, ranging from the medical, to agricultural, and even juridic field. Everyday, use cases where it can help humans process, and assess large volumes of data, in order to learn from it, and make predictions on new data, grow.

Computer vision and applications stemming from it are no different, and ML plays a significant role in them, for instance in self-driving vehicles. Violence detection is another example of a field where the employment of ML could lead to a momentous leap forward, where authorities are remotely notified by a computer, when anomalous / violent activity arises.

The produced document serves as an introduction to ML methods in Violence Detection. State of the Art approaches were covered, and summarized, in an attempt to critically analyse each of them based on their perks, disadvantages. Scenarios in which these yield the best results were also discussed, as well as what methods each of them use in their respective phases. Having covered these, the training methodology that would be followed is detailed, from the architecture of the pipeline, to the used dataset and the pre-processing steps involved in preparing it to be used. After having discussed the training methodology, the models thatt were developed were detailed in an effort to better understand them, and why it's believed that they were suited to detect violent acts in different types of streams. Lastly, the results that were attained from training the proposed models, with the created dataset, in the diverse configurations regarding hyper-parameters are shown and elaborated upon, looking for relations between certains configurations and the effects they had on the different models.

Regarding future work, this stage has several different approaches that can be taken in order improve on the work that was conducted up until the current stage, namely:

1. **Expansion of Dataset** - this is a possible avenue that, if pursued, can lead to a more robustly trained model, which would naturally yield better results, since it was trained more comprehensively. This can be achieved by using the aforementioned dataset that was recorded inside the car and simulates violent acts, or by investigating additional datasets online that can be used to contribute to a more well rounded dataset.

2. **Implement the model in a real time environment** - as has been previously seen in countless other scientific articles, this is a possibility that could be explored in order to leverage the knowledge gained from this Thesis, and apply it in a real life scenario, and have the implemented model predict over data provided from sensors in such scenarios as inside ride sharing vehicles, or other public contexts.

BIBLIOGRAPHY

Almamon Abdali and Rana Al-Tuma. Robust real-time violence detection in video using cnn and lstm. pages 104–108, 03 2019. doi: 10.1109/SCCS.2019.8852616.

Simone Accattoli, Paolo Sernani, Nicola Falcionelli, Dagmawi Neway Mekuria, and Aldo Franco Dragoni. Violence detection in videos by combining 3d convolutional neural networks and support vector machines. *Applied Artificial Intelligence*, 34(4):329–344, 2020.

Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. Multimodal machine learning: A survey and taxonomy. *IEEE transactions on pattern analysis and machine intelligence*, 41(2):423–443, 2018.

Piotr Bilinski and Francois Bremond. Human violence recognition and detection in surveillance videos. In *2016 13th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 30–36. IEEE, 2016.

Castle. What is semi-supervised learning. https://blogs.oracle.com/datascience/what-is-semi-supervised-learning, 2018.

Sarita Chaudhary, Mohd Aamir Khan, and Charul Bhatnagar. Multiple anomalous activity detection in videos. *Procedia Computer Science*, 125:336–345, 2018.

Ankur Datta, Mubarak Shah, and N Da Vitoria Lobo. Person-on-person violence detection in video data. In *Object recognition supported by user interaction for service robots*, volume 1, pages 433–438. IEEE, 2002.

Piotr Dollár, Vincent Rabaud, Garrison Cottrell, and Serge Belongie. *Behavior recognition via sparse spatio-temporal features*. IEEE, 2005.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep learning. book in preparation for mit press. *URL¡ http://www. deeplearningbook. org*, 1, 2016.

Ismael Serrano Gracia, Oscar Deniz Suarez, Gloria Bueno Garcia, and Tae-Kyun Kim. Fast fight detection. *PloS one*, 10(4):e0120448, 2015.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Beth Hoffman and Rupashree Bhattacharya. *Machine Learning, Deep Learning 101*. IBM Developer, 2016. https://developer.ibm.com/articles/l-machine-learning-deep-learning-trs/.

Yossi Itcher. *Real-Time Detection of Violent Crowd Behavior*. Open University of Israel, 2013.

Ru Jia, Ru Li, Meiju Yu, and Shanshan Wang. E-commerce purchase prediction approach by user behavior data. In *2017 international conference on computer, information and telecommunication systems (CITS)*, pages 1–5. IEEE, 2017.

Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.

Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*, 2016.

Bill Kuechler and Vijay Vaishnavi. On theory development in design science research: anatomy of a research project. *European Journal of Information Systems*, 17(5):489–504, 2008.

Yuji Kuno, Takahiro Watanabe, Yoshinori Shimosakoda, and Satoshi Nakagawa. Automated detection of human for visual surveillance system. In *Proceedings of 13th International Conference on Pattern Recognition*, volume 3, pages 865–869. IEEE, 1996.

Ji Li, Xinghao Jiang, Tanfeng Sun, and Ke Xu. Efficient violence detection using 3d convolutional neural networks. In *2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, 2019.

Tom M Mitchell et al. Machine learning. 1997. *Burr Ridge, IL: McGraw Hill*, 45(37):870–877, 1997.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.

Enrique Bermejo Nievas, Oscar Deniz Suarez, Gloria Bueno García, and Rahul Sukthankar. Violence detection in video using computer vision techniques. In *International conference on Computer analysis of images and patterns*, pages 332–339. Springer, 2011.

Ken Peffers, Tuure Tuunanen, Marcus A Rothenberger, and Samir Chatterjee. A design science research methodology for information systems research. *Journal of management information systems*, 24(3):45–77, 2007.

Bruno Malveira Peixoto, Sandra Avila, Zanoni Dias, and Anderson Rocha. Breaking down violence: A deep-learning strategy to model and classify violence in videos. In *Proceedings of the 13th International Conference on Availability, Reliability and Security*, pages 1–7, 2018.

Florent Perronnin, Jorge Sánchez, and Thomas Mensink. Improving the fisher kernel for large-scale image classification. In *European conference on computer vision*, pages 143–156. Springer, 2010.

Muhammad Ramzan, Adnan Abid, Hikmat Ullah Khan, Shahid Mahmood Awan, Amina Ismail, Muzamil Ahmed, Mahwish Ilyas, and Ahsan Mahmood. A review on state-of-the-art violence detection techniques. *IEEE Access*, 7:107560–107575, 2019.

Kishore K Reddy and Mubarak Shah. Recognizing 50 human action categories of web videos. *Machine vision and applications*, 24(5):971–981, 2013.

Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

Ismael Serrano, Oscar Deniz, Jose Luis Espinosa-Aranda, and Gloria Bueno. Fight recognition in video using hough forests and 2d convolutional neural network. *IEEE Transactions on Image Processing*, 27(10):4787–4797, 2018.

Saman Siadati. *What is UnSupervised Learning?* Strategic Intelligence Research Lab (SIR-LAB), 2018. https://www.researchgate.net/publication/342121950_What_is_UnSupervised_Learning.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Cees GM Snoek, Marcel Worring, and Arnold WM Smeulders. Early versus late fusion in semantic video analysis. In *Proceedings of the 13th annual ACM international conference on Multimedia*, pages 399–402, 2005.

Mohamed Mostafa Soliman, Mohamed Hussein Kamal, Mina Abd El-Massih Nashed, Youssef Mohamed Mostafa, Bassel Safwat Chawky, and Dina Khattab. Violence recognition from videos using deep learning techniques. In *2019 Ninth International Conference on Intelligent Computing and Information Systems (ICICIS)*, pages 80–85. IEEE, 2019.

Xiaoyu Song, Hong Chen, Qing Wang, Yunqiang Chen, Mengxiao Tian, and Hui Tang. A review of audio-visual fusion with machine learning. In *Journal of Physics: Conference Series*, volume 1237, page 022144. IOP Publishing, 2019.

Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

Swathikiran Sudhakaran and Oswald Lanz. Learning to detect violent videos using convolutional long short-term memory. In *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6. IEEE, 2017.

Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018.

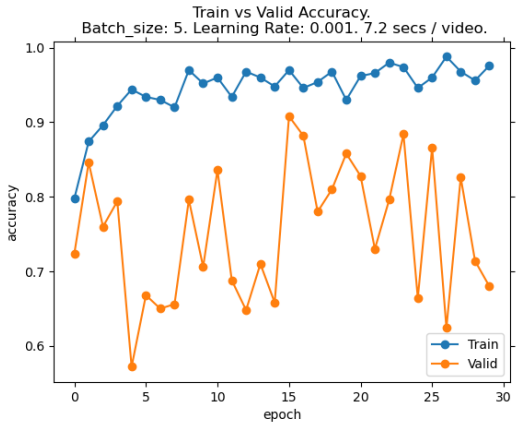
Fath U Min Ullah, Amin Ullah, Khan Muhammad, Ijaz Ul Haq, and Sung Wook Baik. Violence detection using spatiotemporal features with 3d convolutional neural network. *Sensors*, 19(11):2472, 2019.

M. Worring, C. G. M. Snoek, O. de Rooij, G. P. Nguyen, and A. W. M. Smeulders. The mediamill semantic video search engine. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*, volume 4, pages IV–1213–IV–1216, 2007. doi: 10.1109/ICASSP.2007.367294.

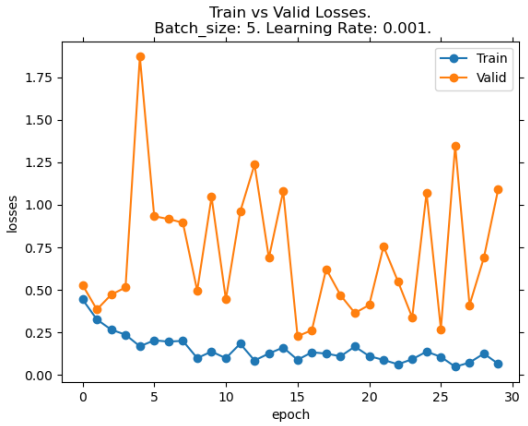
Jianbin Xie, Wei Yan, Chundi Mu, Tong Liu, Peiqin Li, and Shuicheng Yan. Recognizing violent activity without decoding video streams. *Optik*, 127(2):795–801, 2016.

Part I

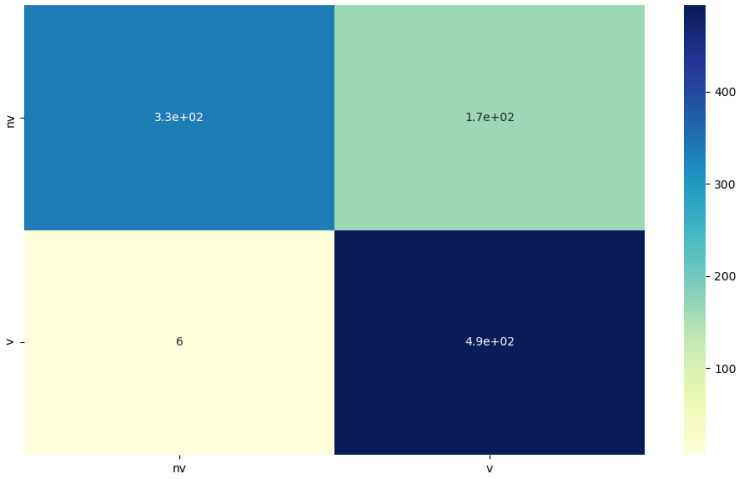
APPENDICES



(a) Accuracy

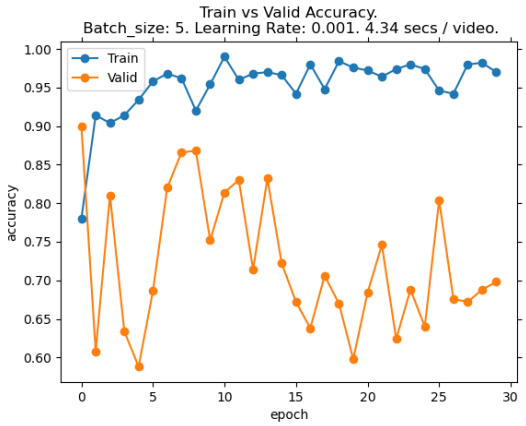


(b) Loss

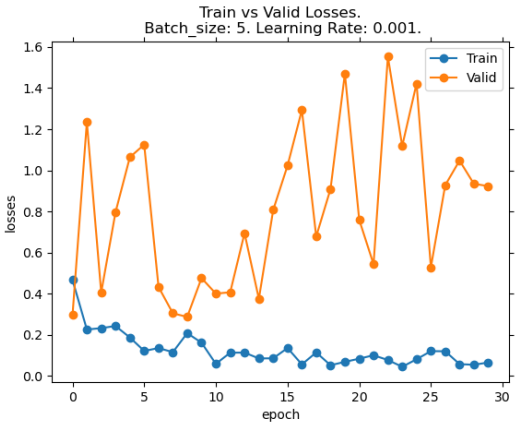


(c) Confusion Matrix

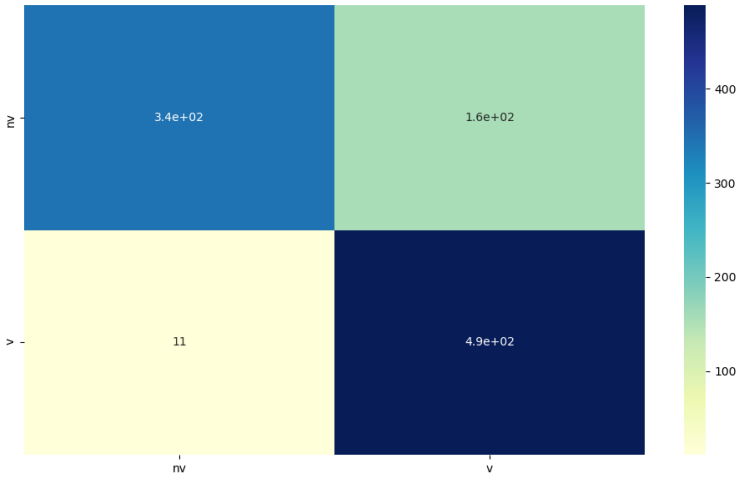
Figure 7: Resnet18's Statistics



(a) Accuracy

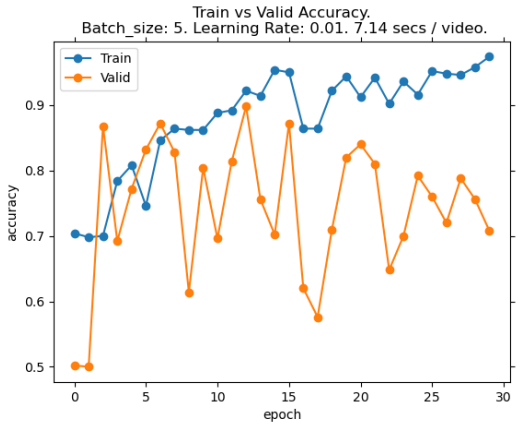


(b) Loss

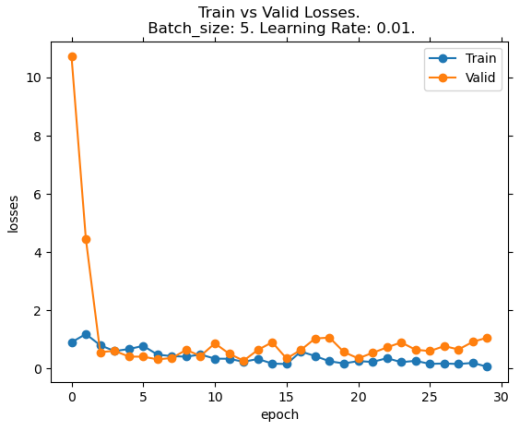


(c) Confusion Matrix

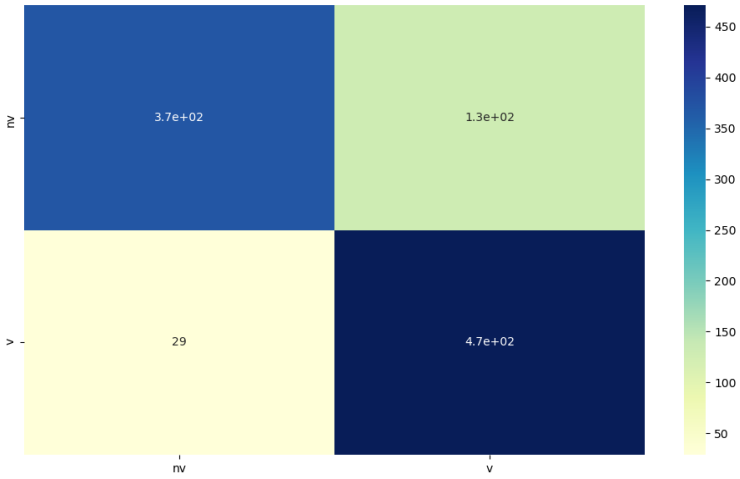
Figure 8: Resnet34's Statistics



(a) Accuracy

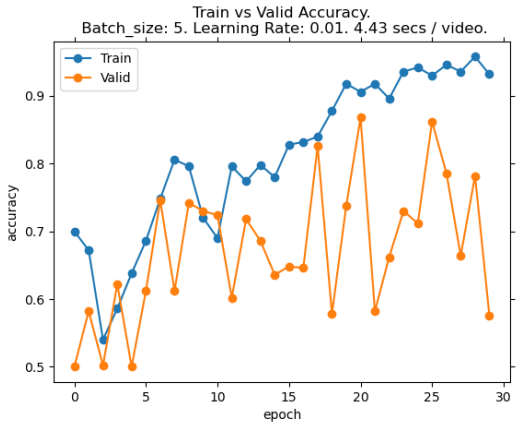


(b) Loss

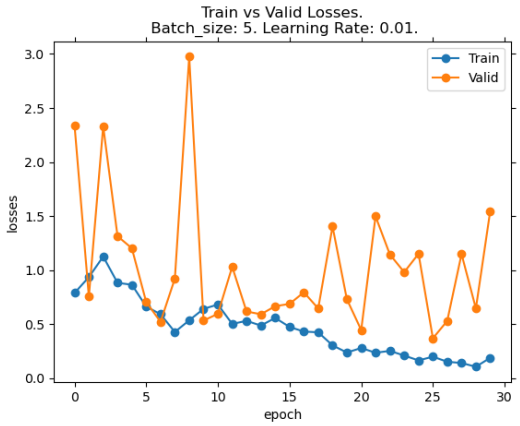


(c) Confusion Matrix

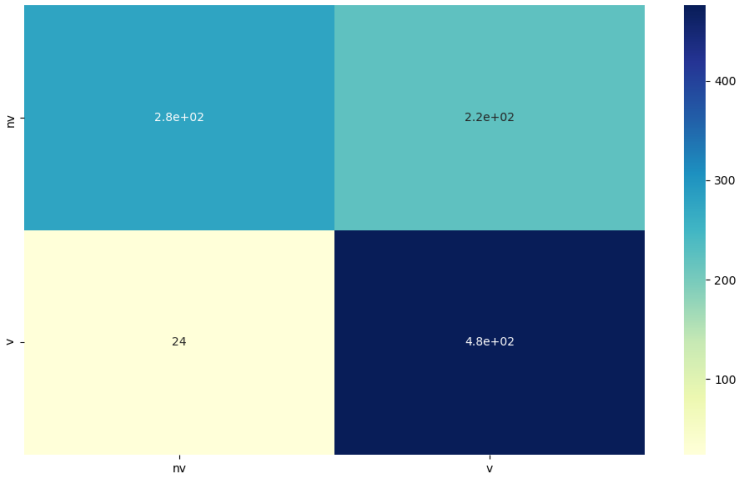
Figure 9: Resnet18's Statistics



(a) Accuracy

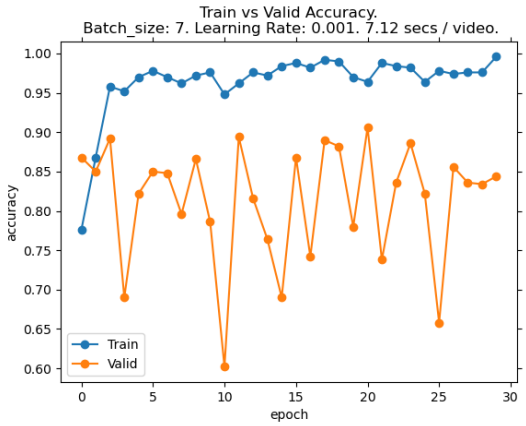


(b) Loss

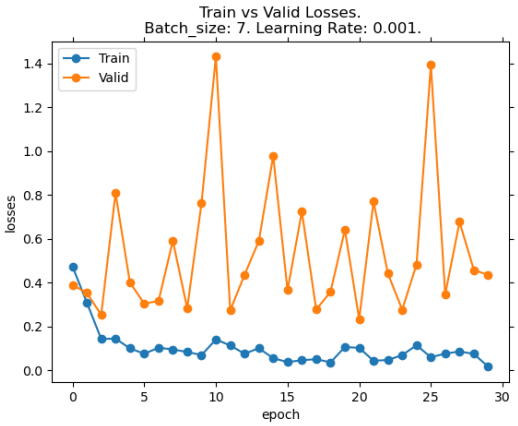


(c) Confusion Matrix

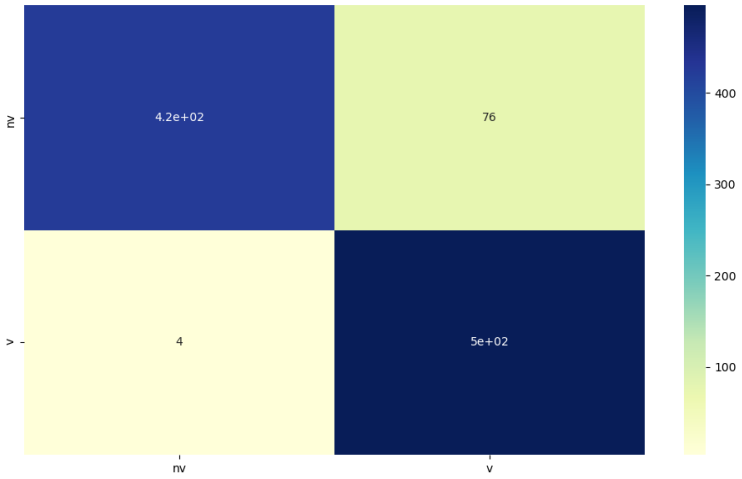
Figure 10: Resnet34's Statistics



(a) Accuracy

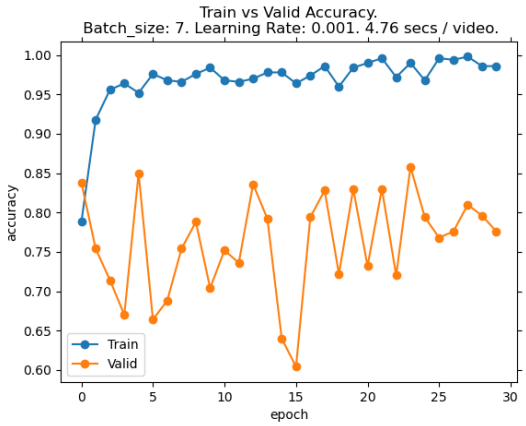


(b) Loss

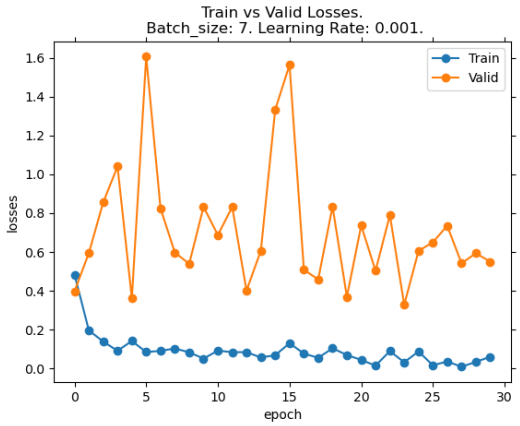


(c) Confusion Matrix

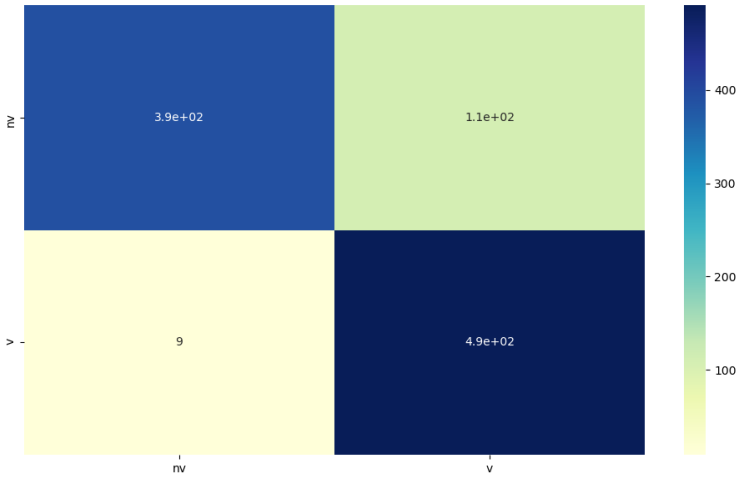
Figure 11: Resnet18's Statistics



(a) Accuracy

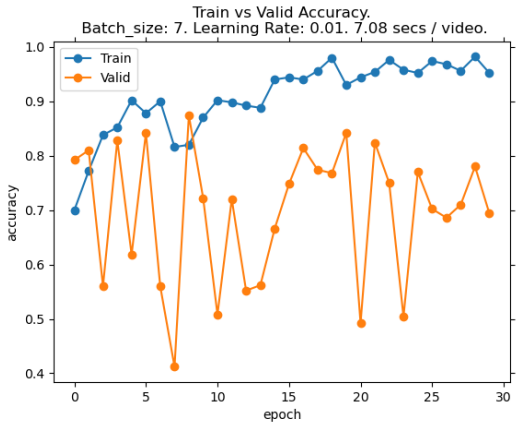


(b) Loss

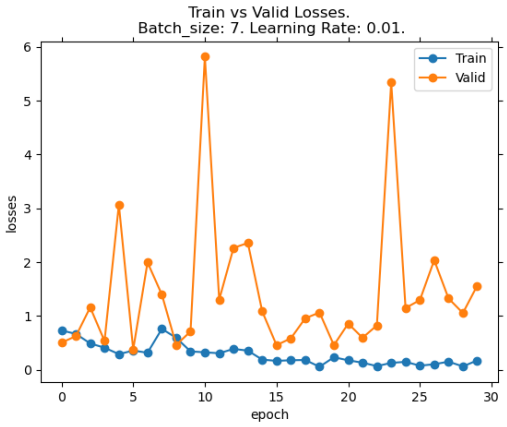


(c) Confusion Matrix

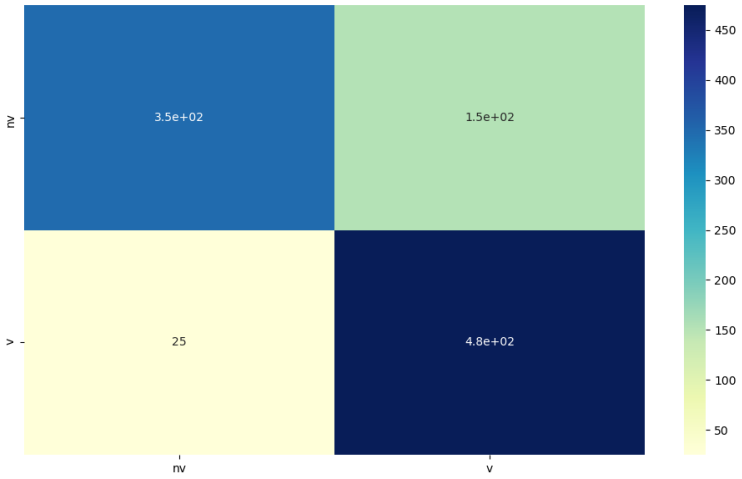
Figure 12: Resnet34's Statistics



(a) Accuracy

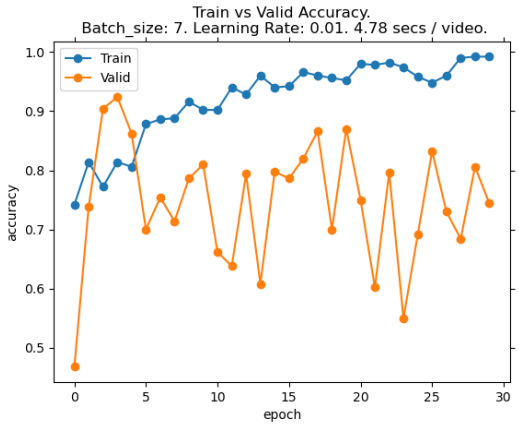


(b) Loss

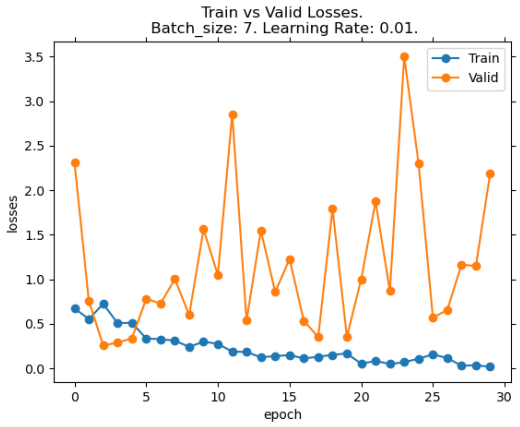


(c) Confusion Matrix

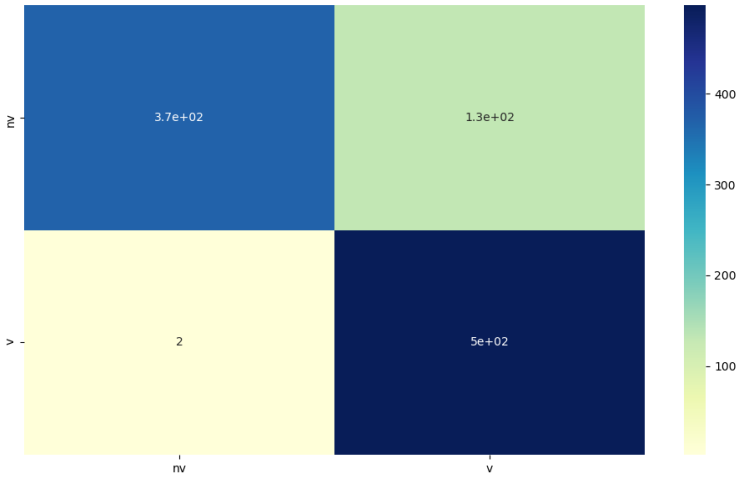
Figure 13: Resnet18's Statistics



(a) Accuracy

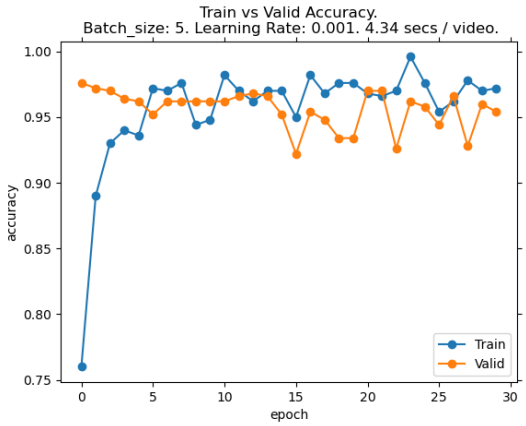


(b) Loss

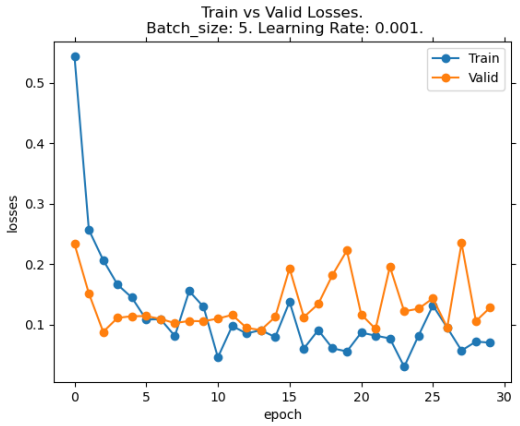


(c) Confusion Matrix

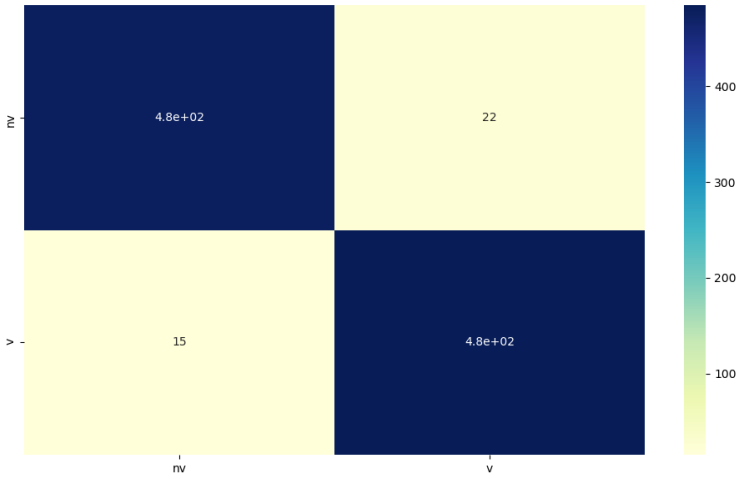
Figure 14: Resnet34's Statistics



(a) Accuracy

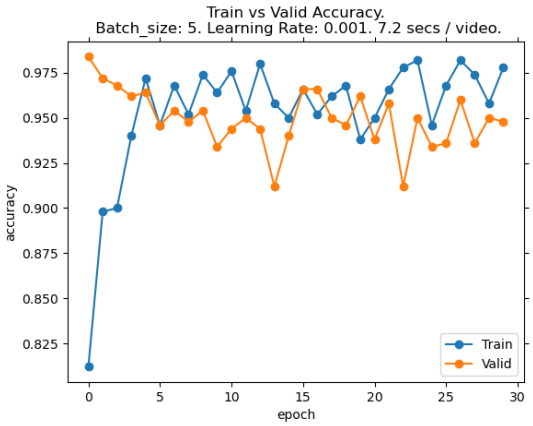


(b) Loss

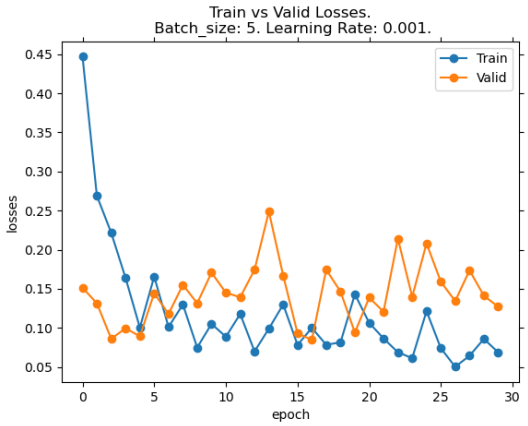


(c) Confusion Matrix

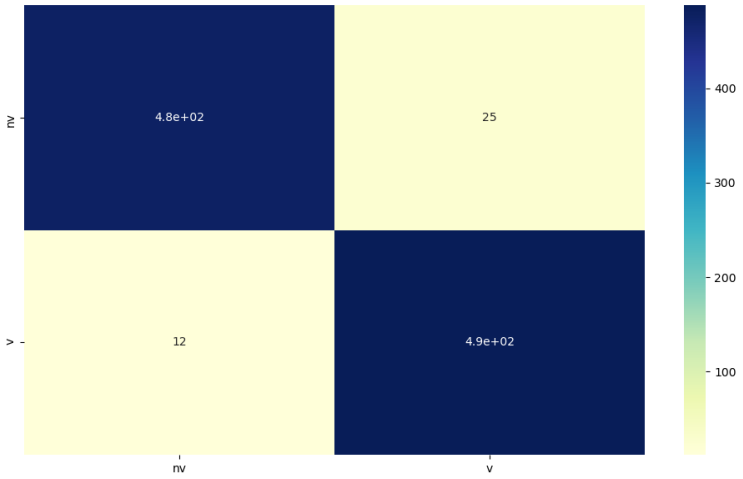
Figure 15: Resnet_MC18's Statistics



(a) Accuracy

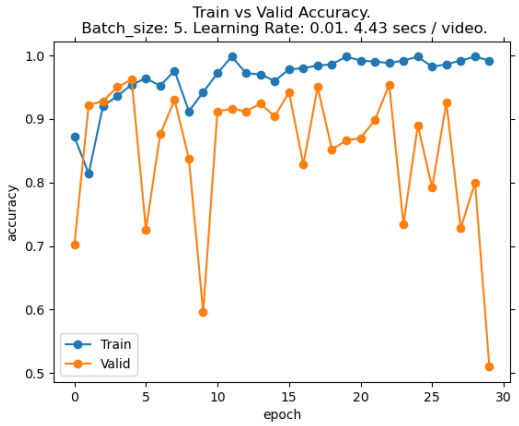


(b) Loss

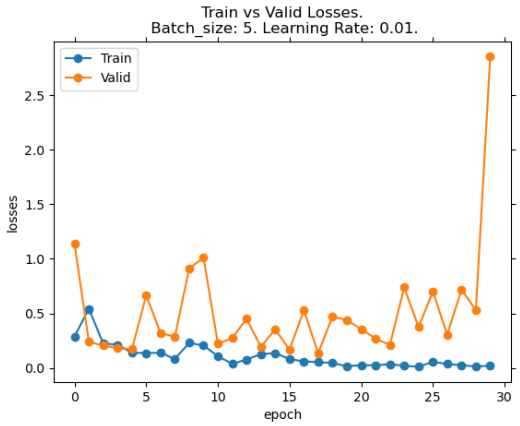


(c) Confusion Matrix

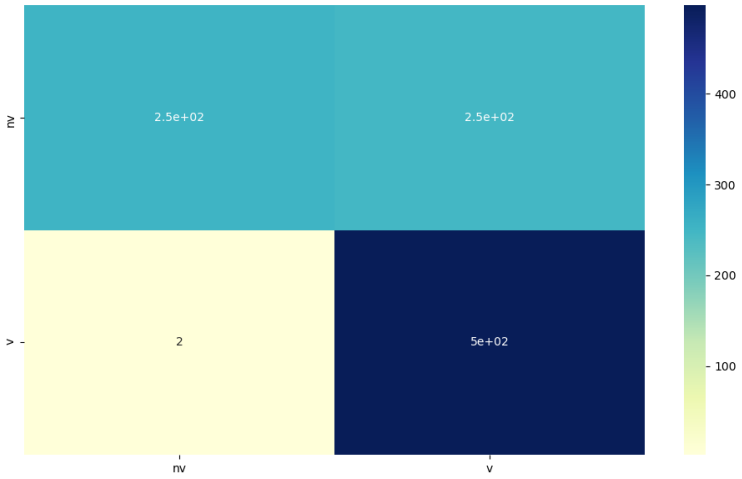
Figure 16: Resnet(2+1)D's Statistics



(a) Accuracy

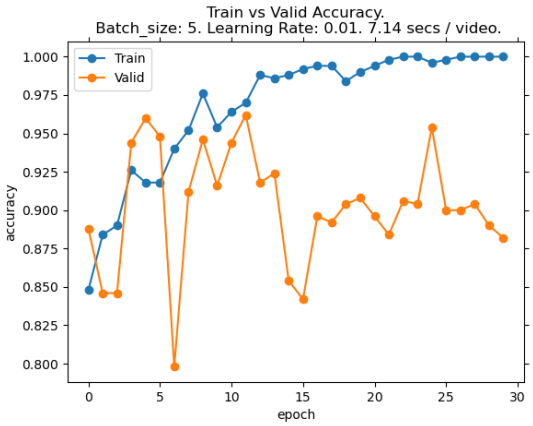


(b) Loss

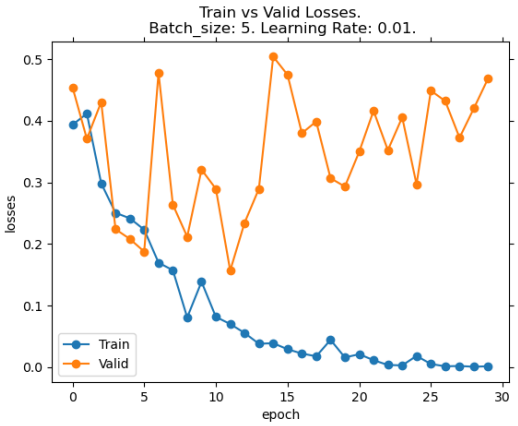


(c) Confusion Matrix

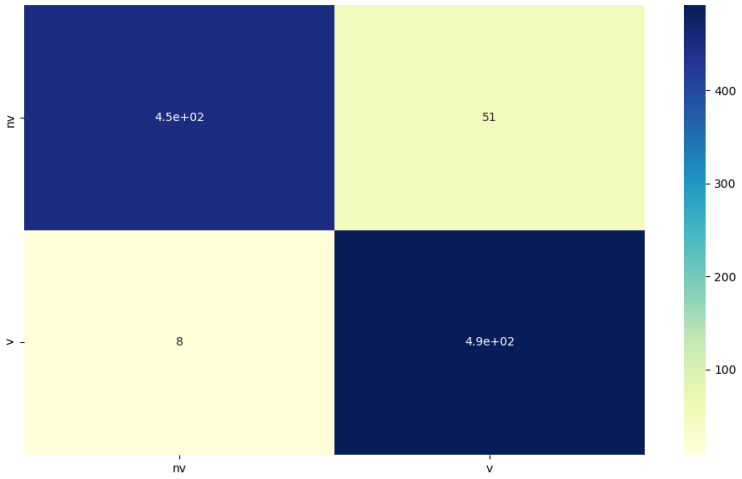
Figure 17: Resnet_MC18's Statistics



(a) Accuracy

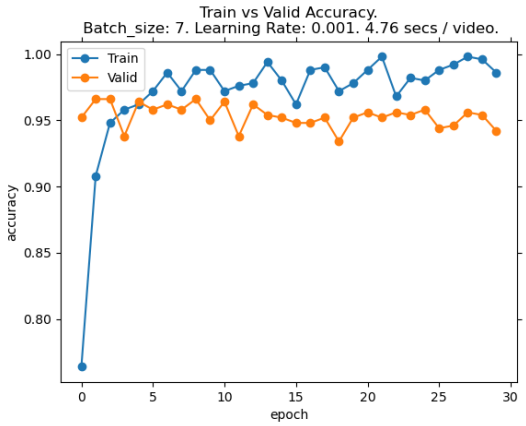


(b) Loss

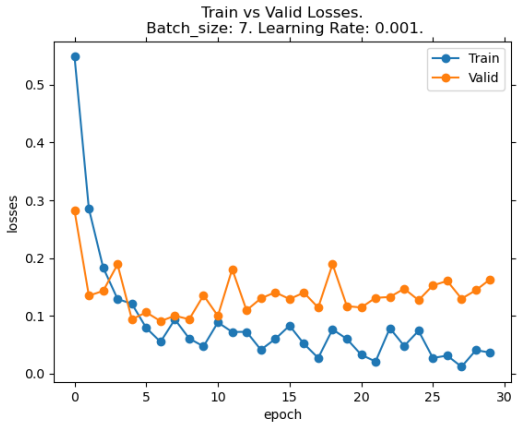


(c) Confusion Matrix

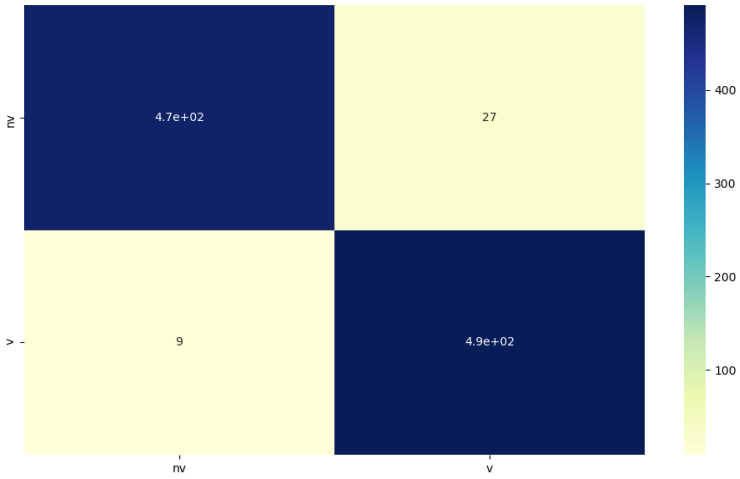
Figure 18: Resnet(2+1)D's Statistics



(a) Accuracy

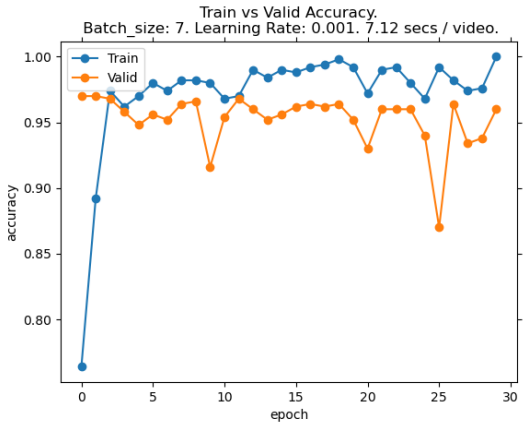


(b) Loss

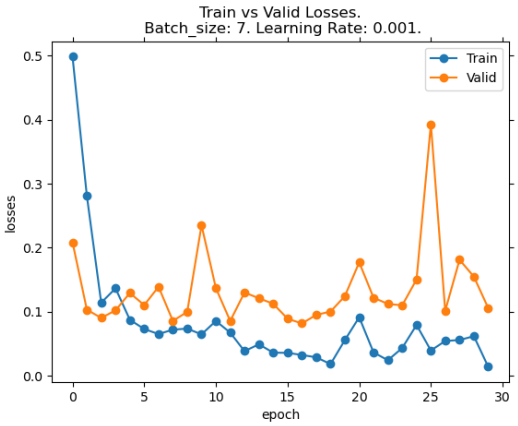


(c) Confusion Matrix

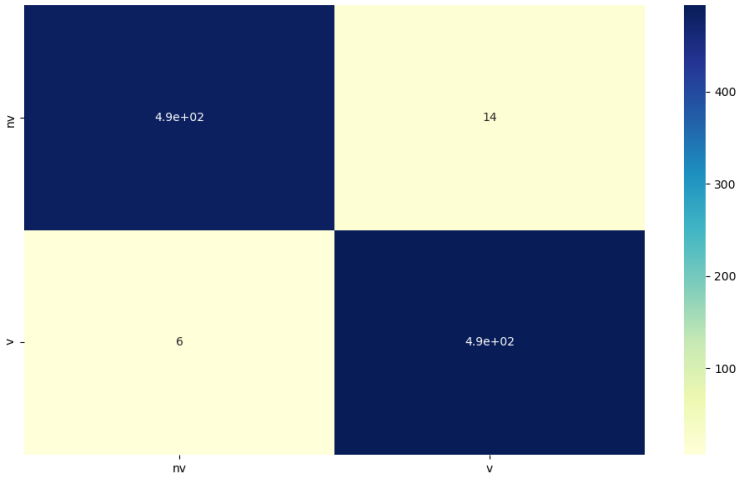
Figure 19: Resnet_MC18's Statistics



(a) Accuracy

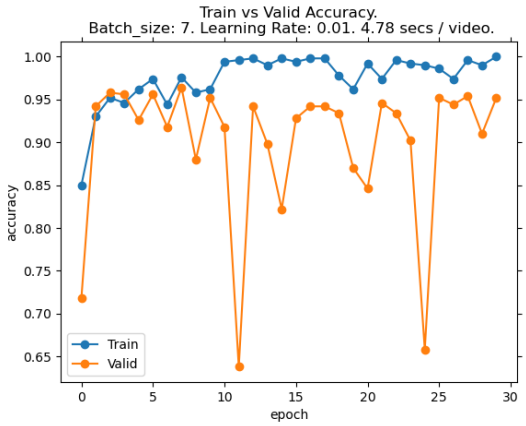


(b) Loss

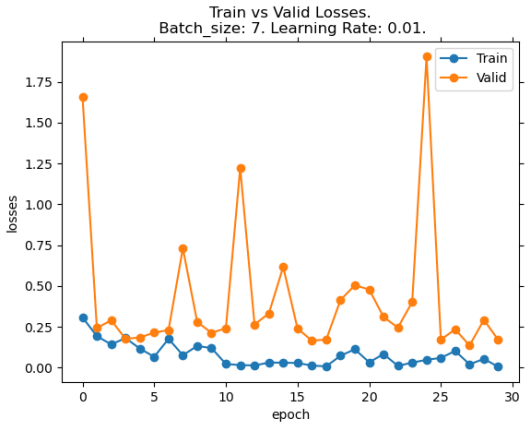


(c) Confusion Matrix

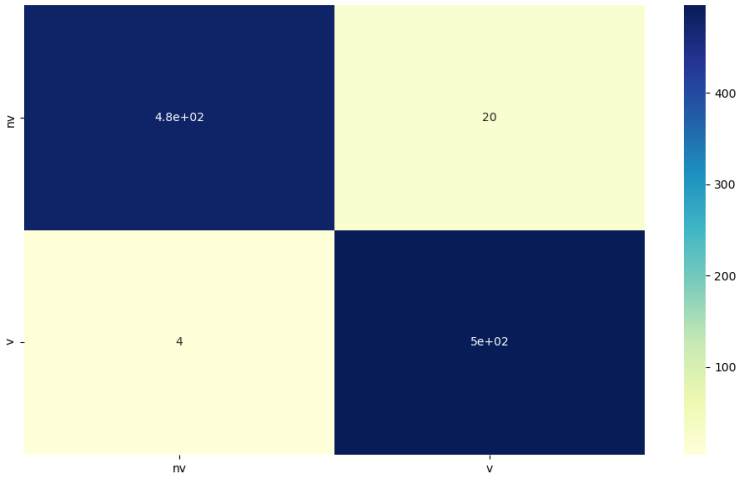
Figure 20: Resnet(2+1)D's Statistics



(a) Accuracy

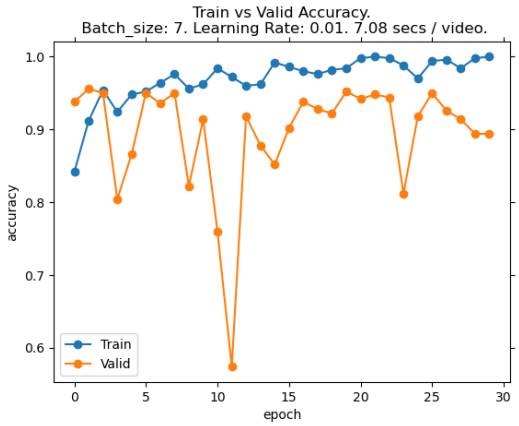


(b) Loss

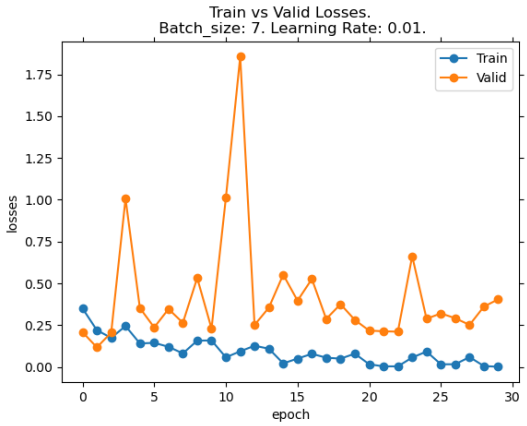


(c) Confusion Matrix

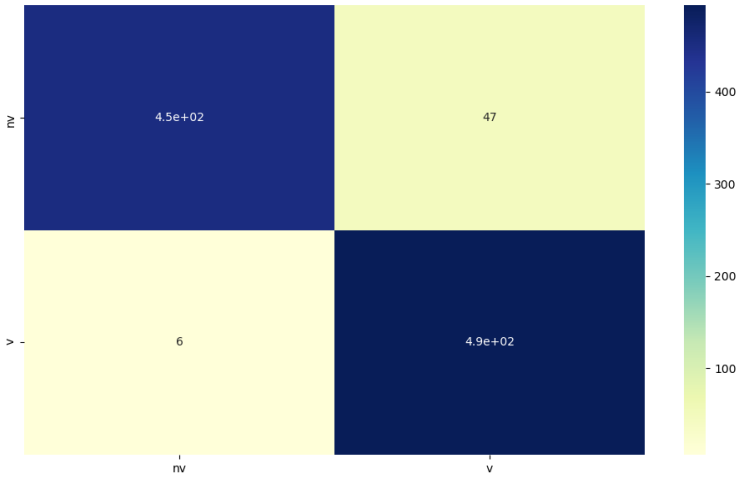
Figure 21: Resnet_MC18's Statistics



(a) Accuracy

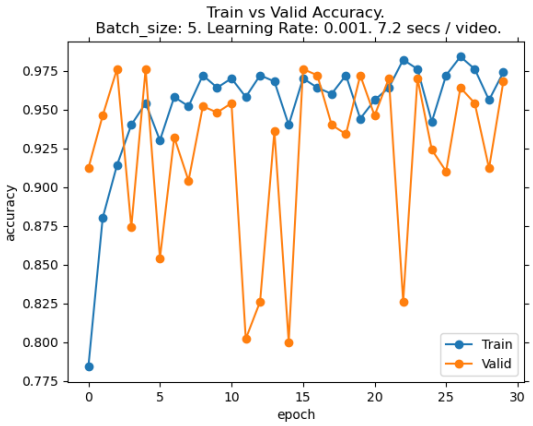


(b) Loss

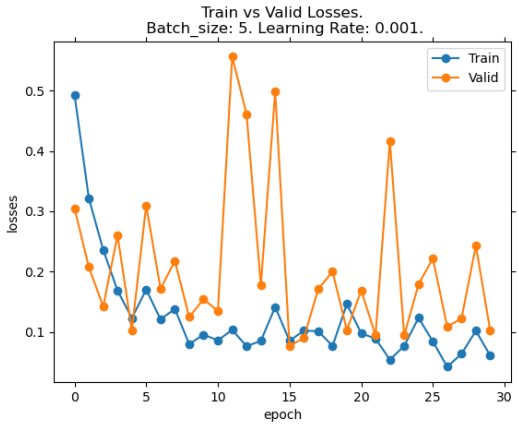


(c) Confusion Matrix

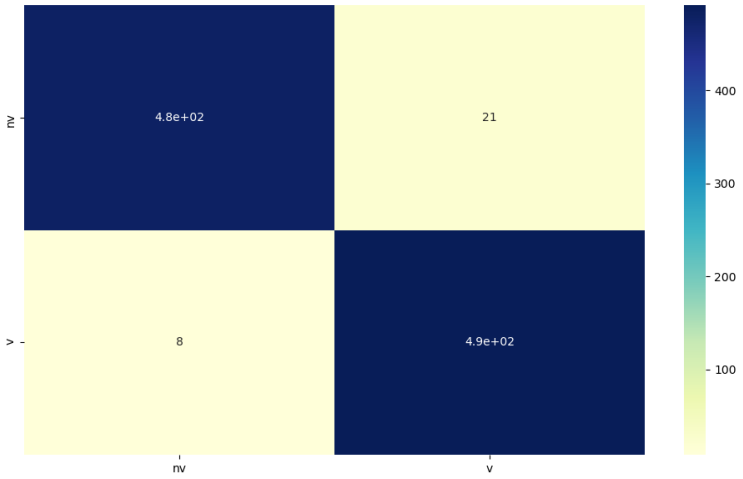
Figure 22: Resnet(2+1)D's Statistics



(a) Accuracy

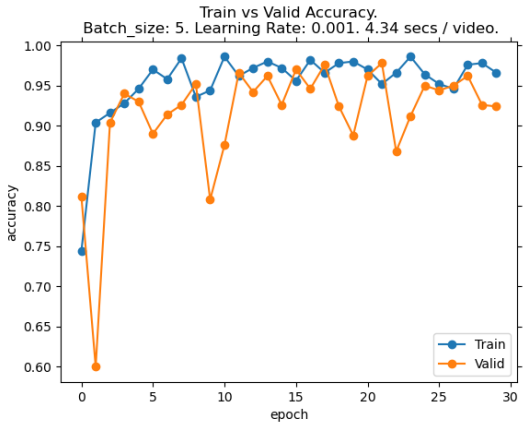


(b) Loss

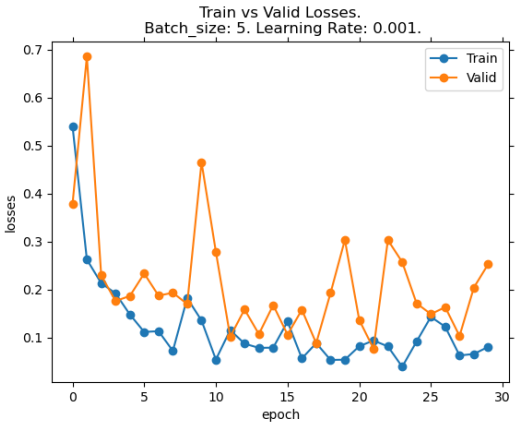


(c) Confusion Matrix

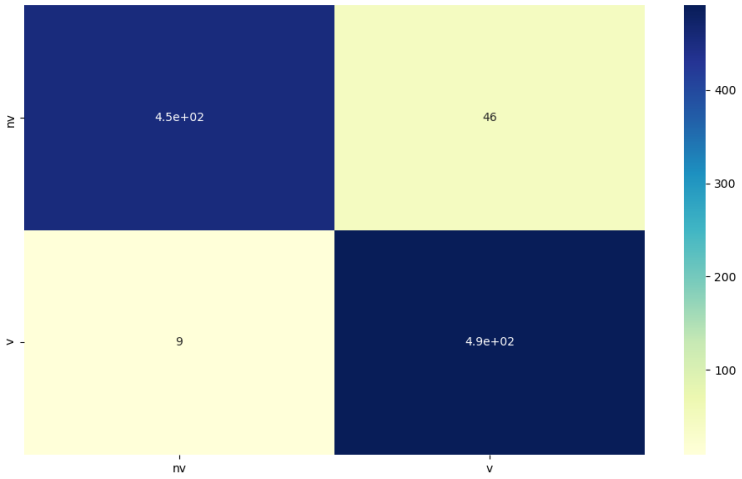
Figure 23: Resnet18 and Resnet(2+1)D's Statistics



(a) Accuracy

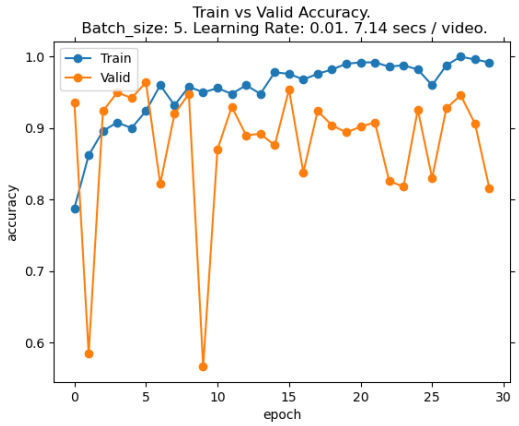


(b) Loss

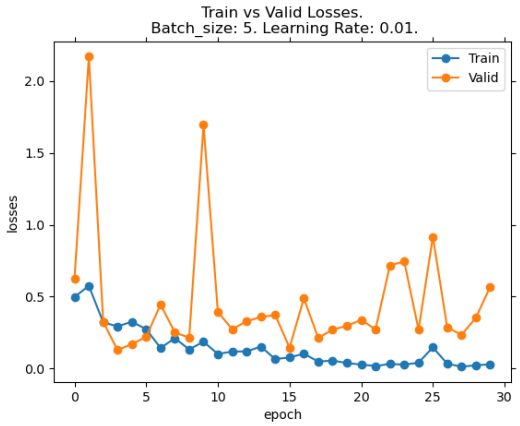


(c) Confusion Matrix

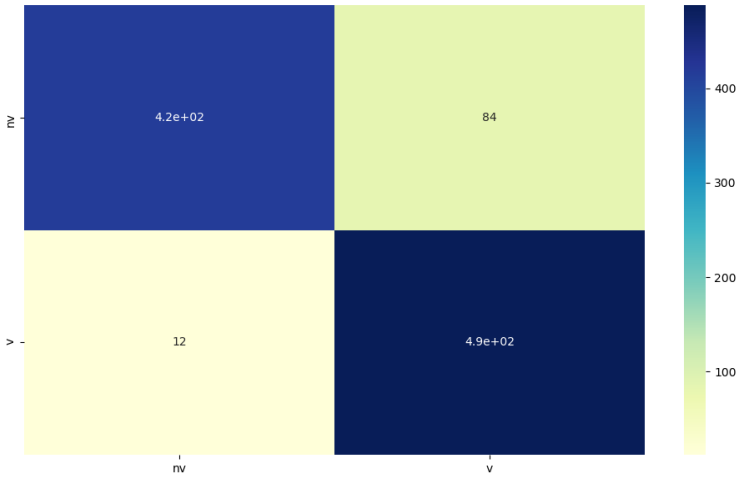
Figure 24: Resnet34 and Resnet MC18's Statistics



(a) Accuracy

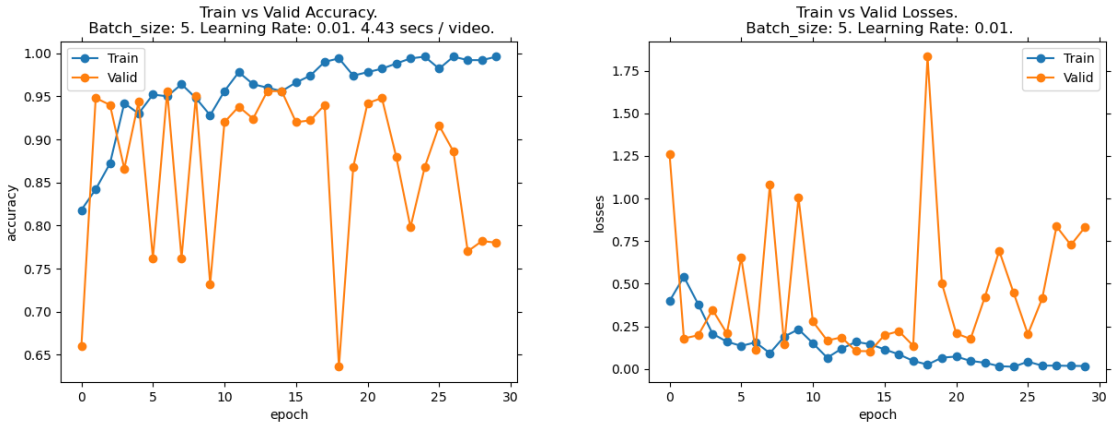


(b) Loss



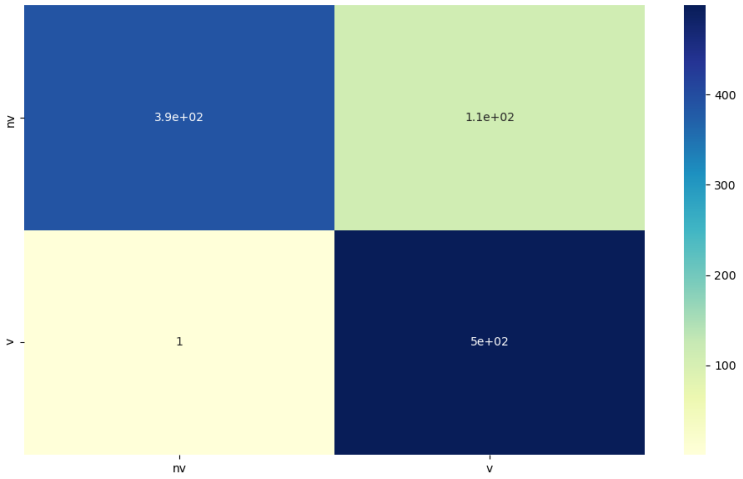
(c) Confusion Matrix

Figure 25: Resnet18 and Resnet(2+1)D's Statistics



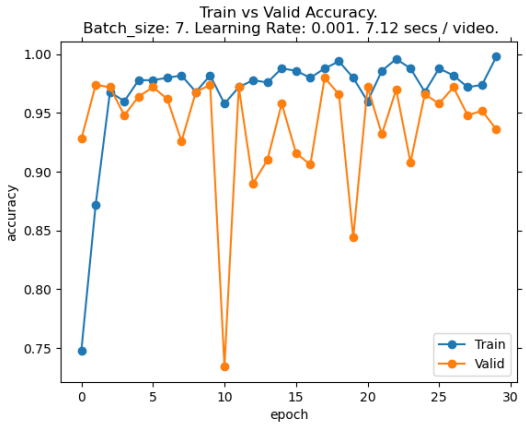
(a) Accuracy

(b) Loss

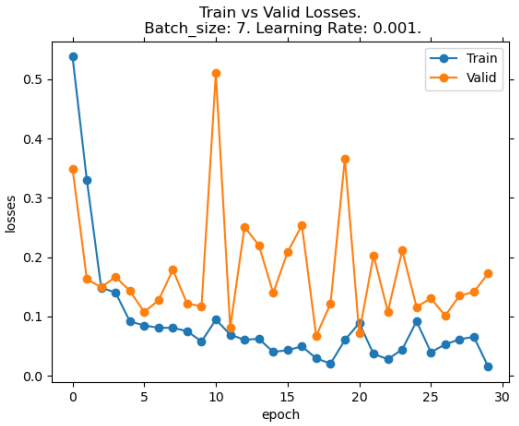


(c) Confusion Matrix

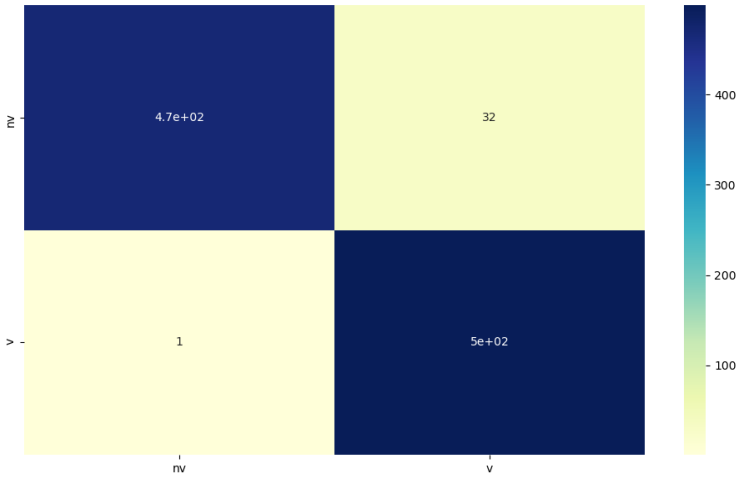
Figure 26: Resnet34 and Resnet MC18's Statistics



(a) Accuracy

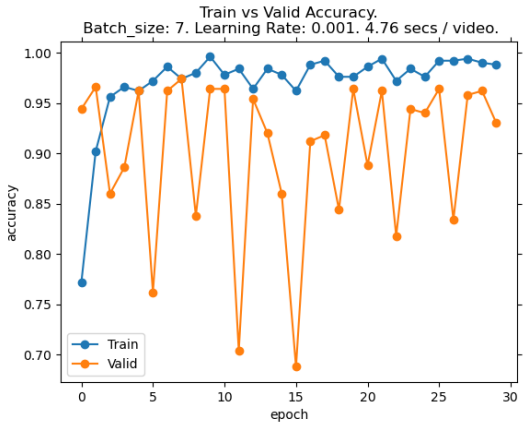


(b) Loss

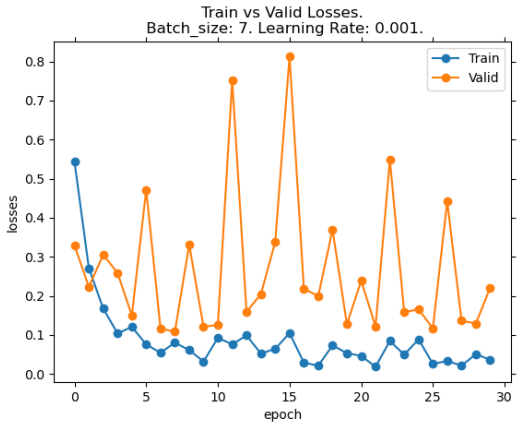


(c) Confusion Matrix

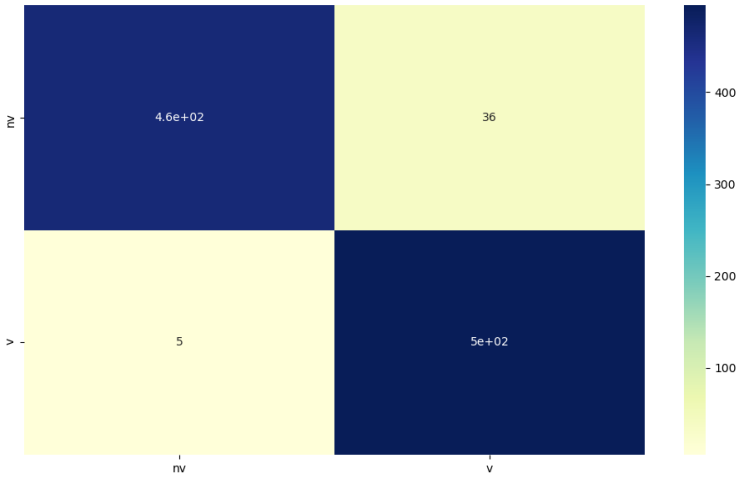
Figure 27: Resnet18 and Resnet(2+1)D's Statistics



(a) Accuracy

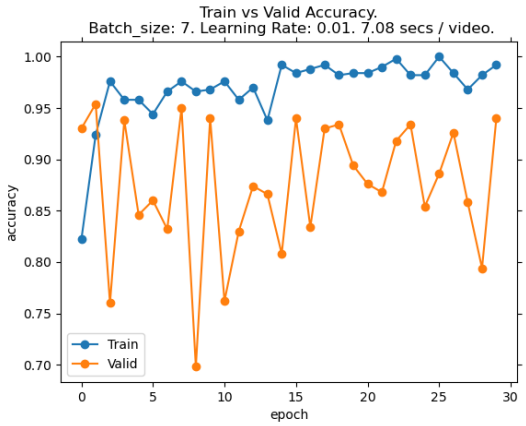


(b) Loss

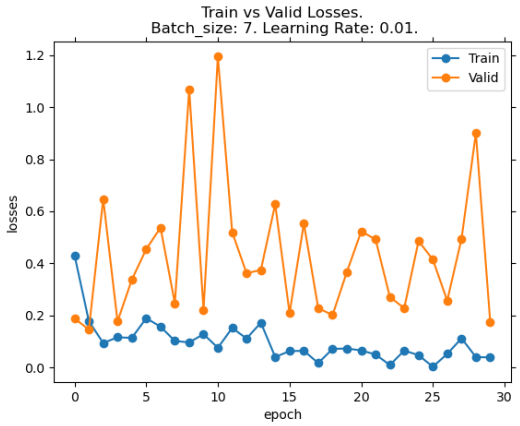


(c) Confusion Matrix

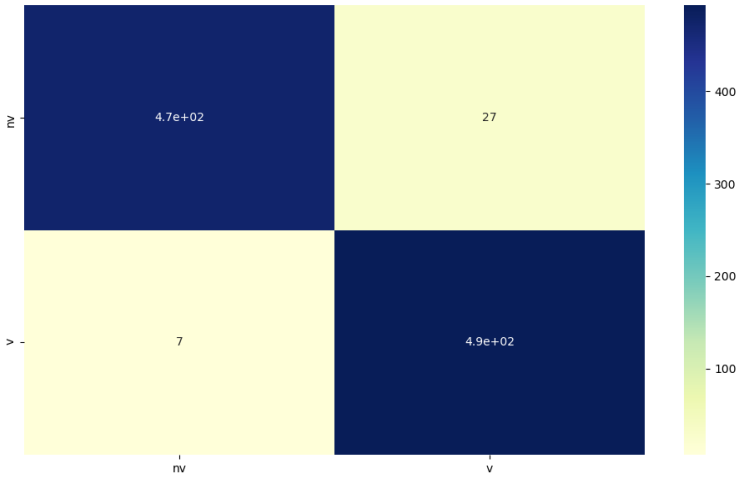
Figure 28: Resnet34 and Resnet MC18's Statistics



(a) Accuracy

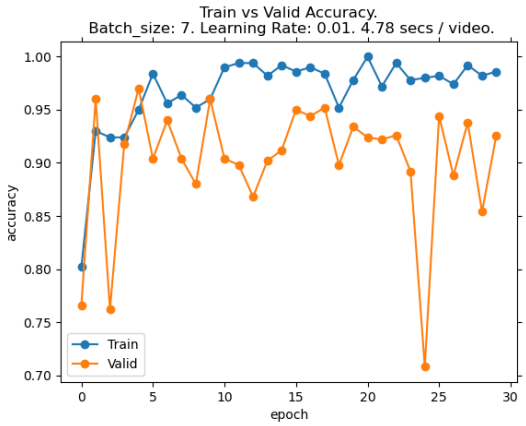


(b) Loss

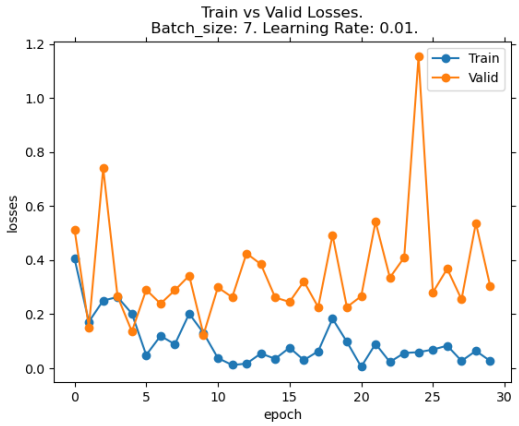


(c) Confusion Matrix

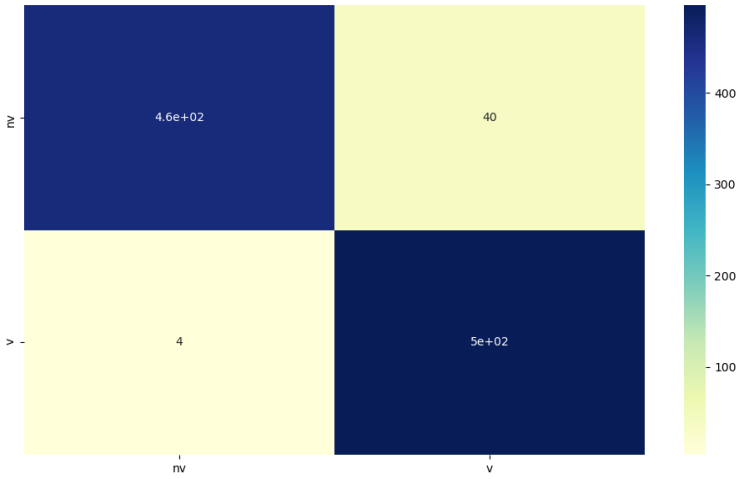
Figure 29: Resnet18 and Resnet(2+1)D's Statistics



(a) Accuracy

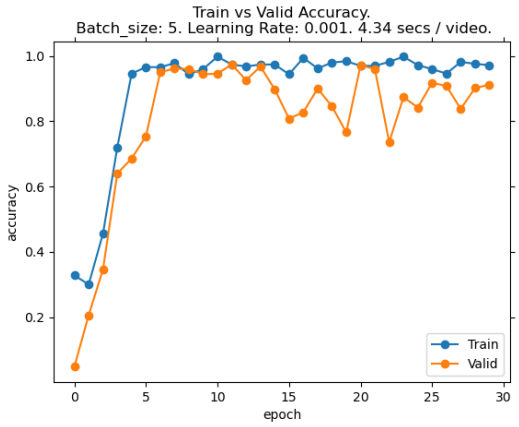


(b) Loss

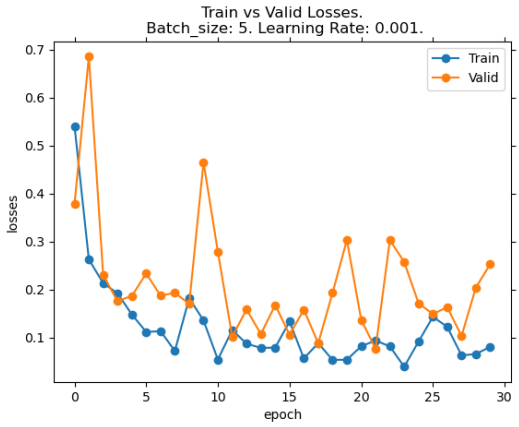


(c) Confusion Matrix

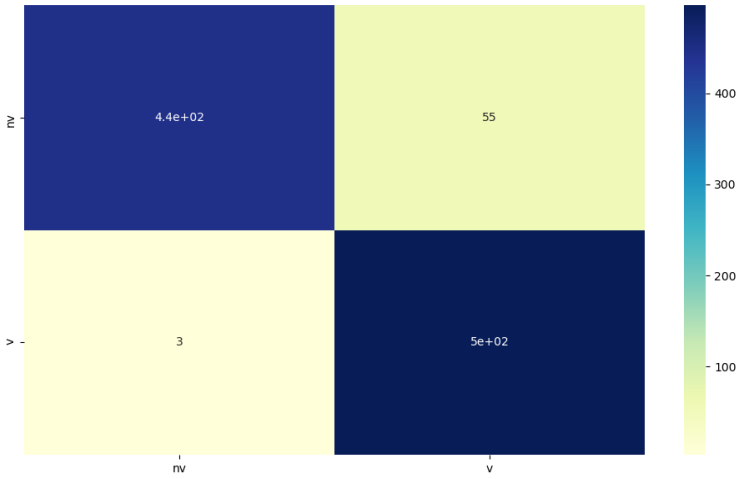
Figure 30: Resnet34 and Resnet MC18's Statistics



(a) Accuracy

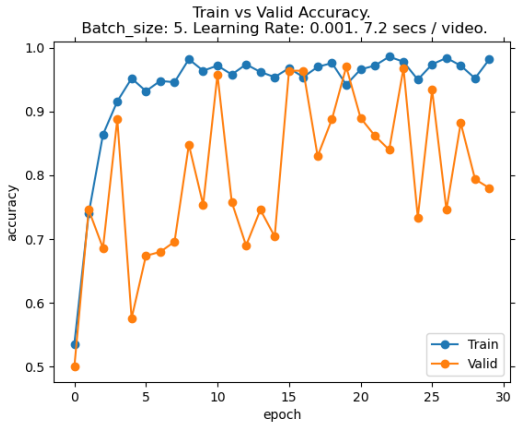


(b) Loss

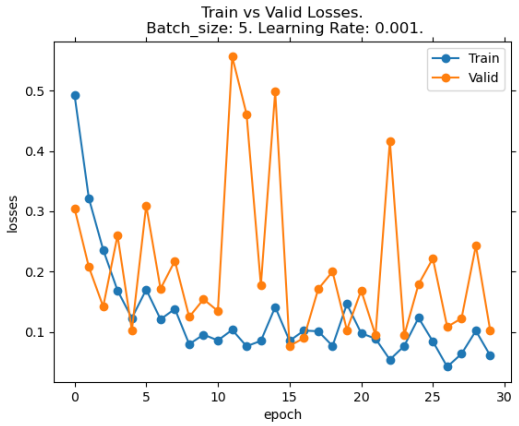


(c) Confusion Matrix

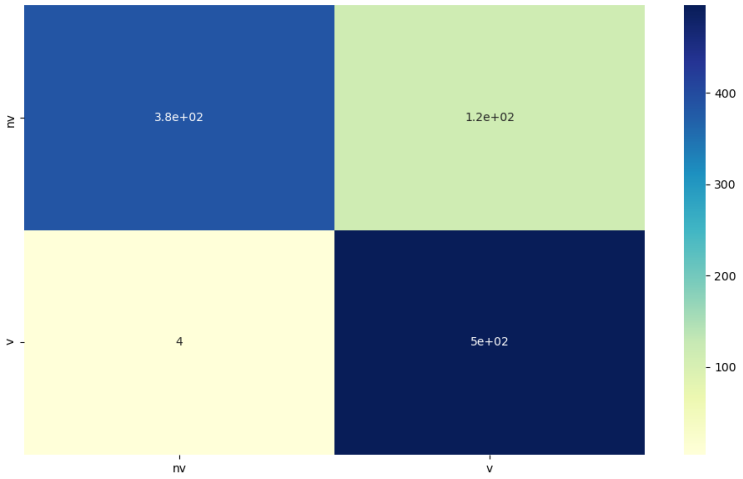
Figure 31: Late Fusion Model's Statistics



(a) Accuracy

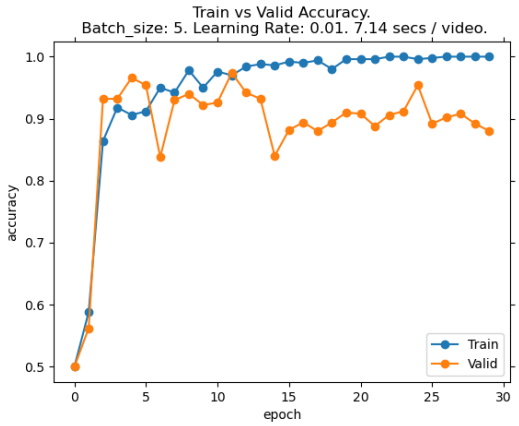


(b) Loss

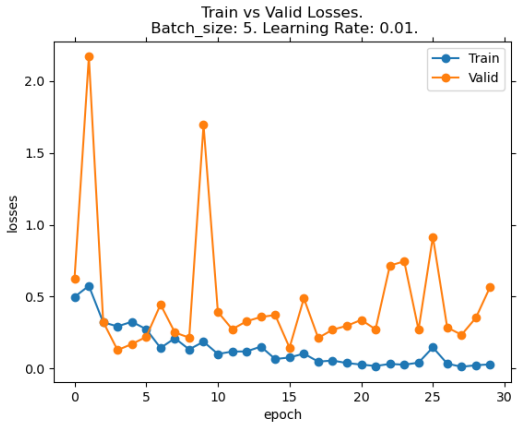


(c) Confusion Matrix

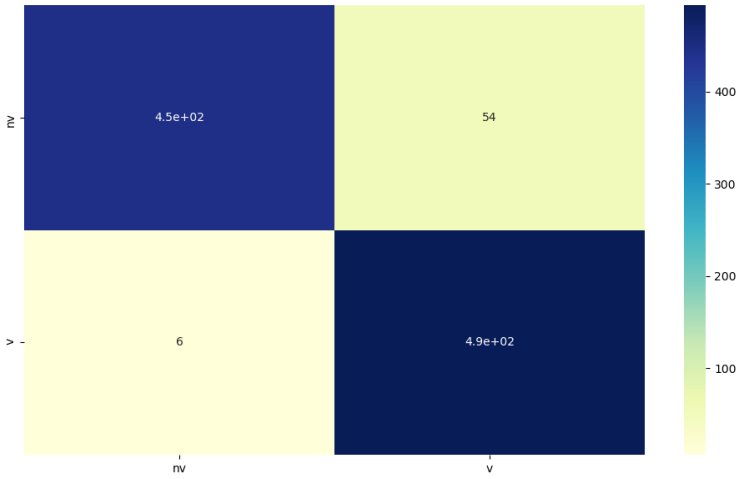
Figure 32: Late Fusion Model's Statistics



(a) Accuracy

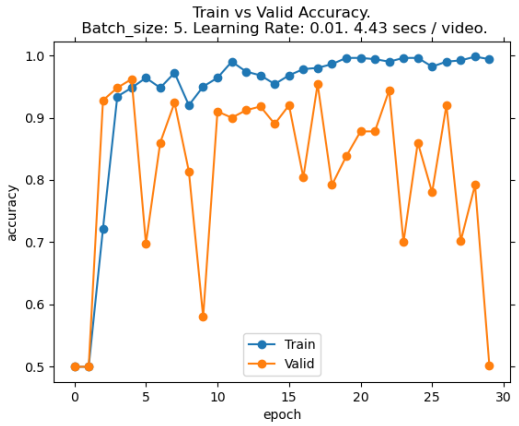


(b) Loss

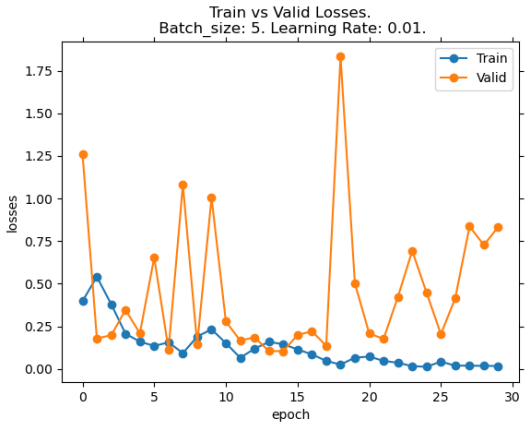


(c) Confusion Matrix

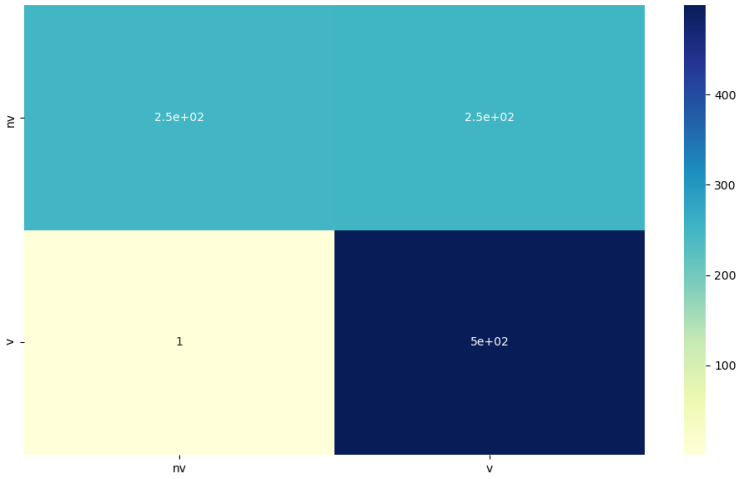
Figure 33: Late Fusion Model's Statistics



(a) Accuracy

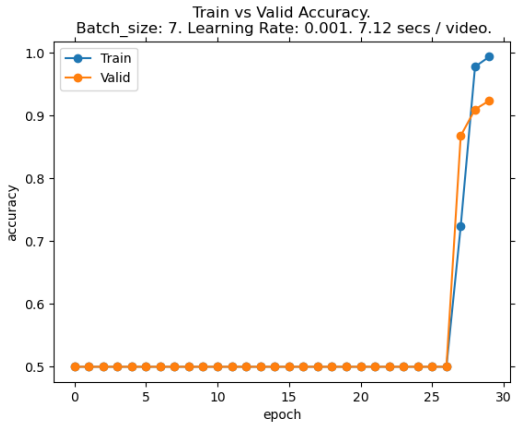


(b) Loss

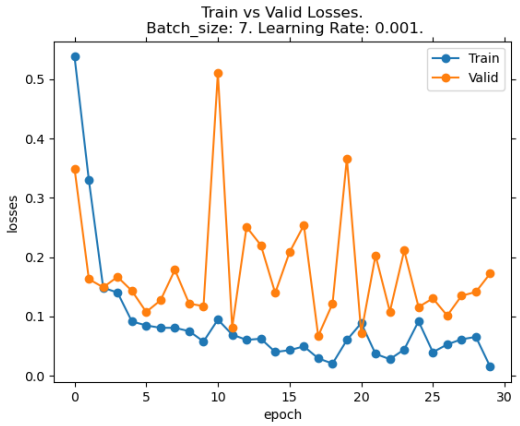


(c) Confusion Matrix

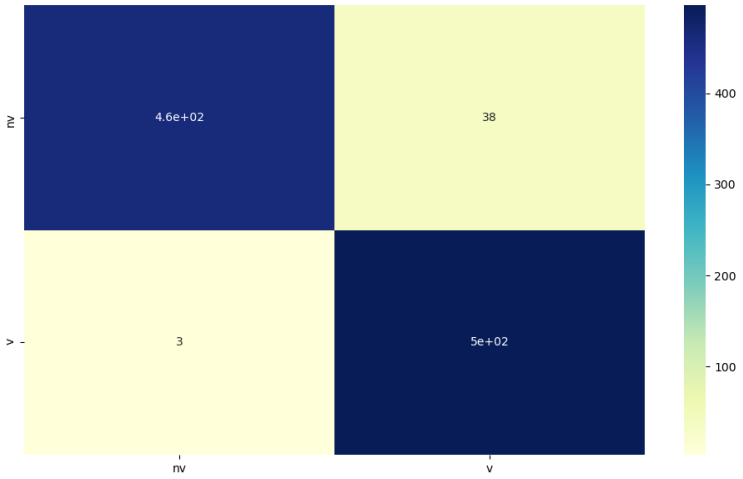
Figure 34: Late Fusion Model's Statistics



(a) Accuracy

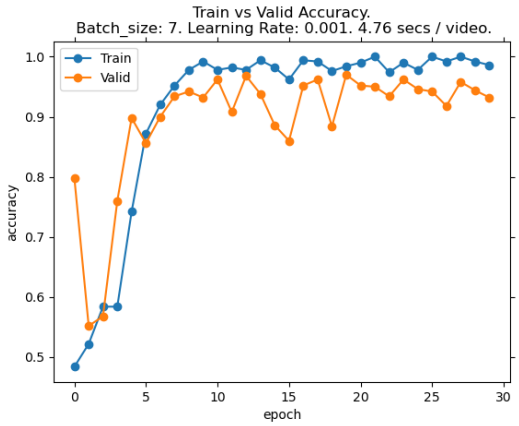


(b) Loss

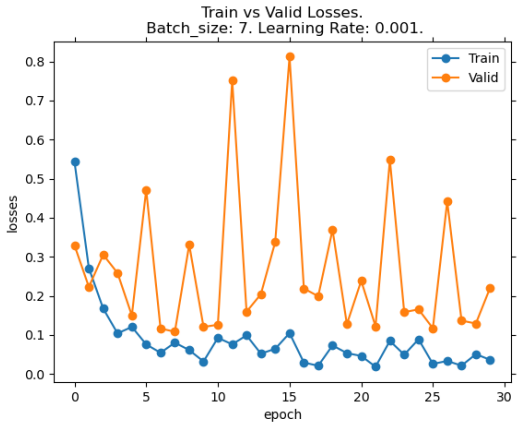


(c) Confusion Matrix

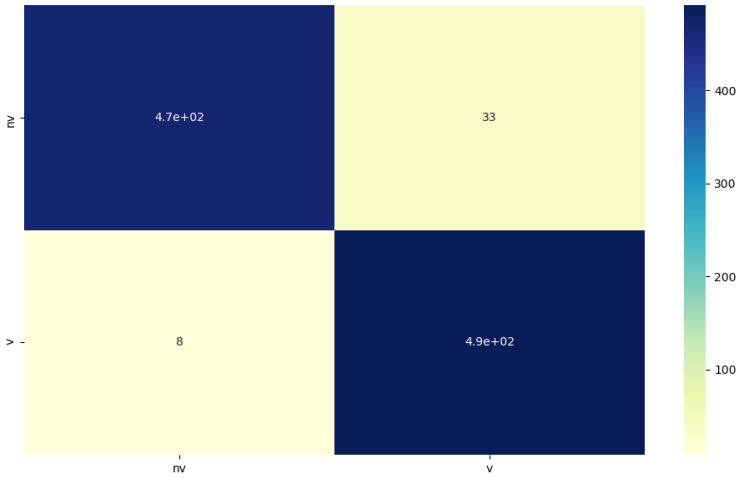
Figure 35: Late Fusion Model's Statistics



(a) Accuracy

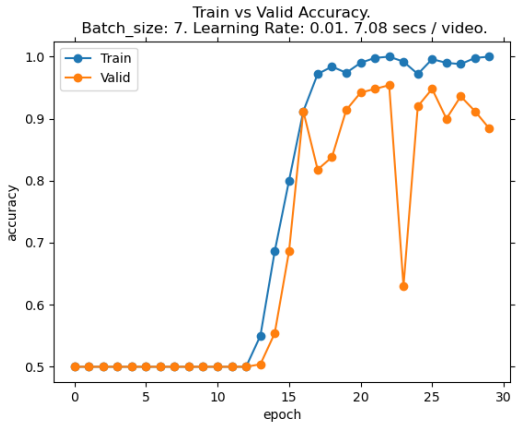


(b) Loss

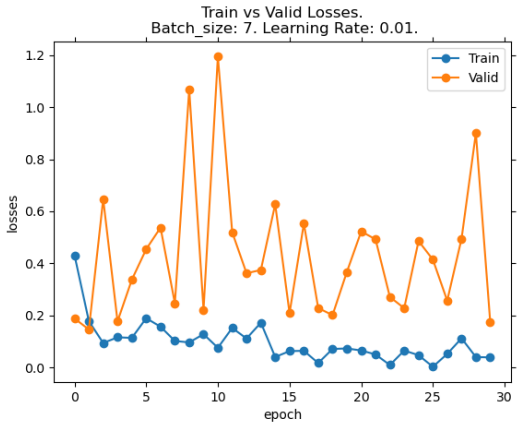


(c) Confusion Matrix

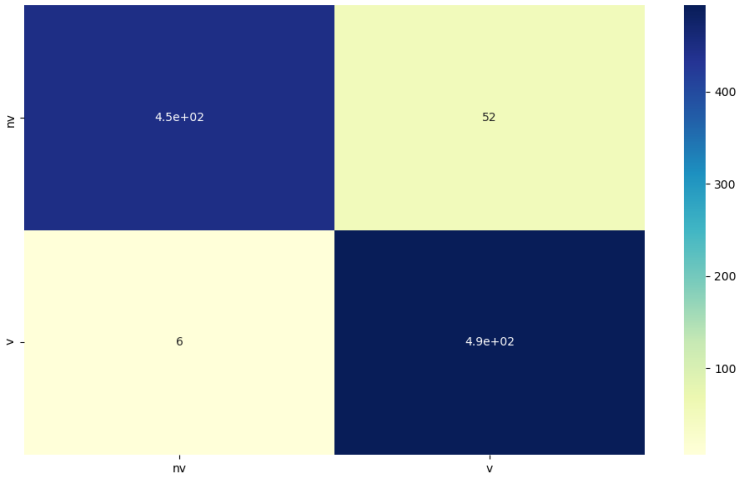
Figure 36: Late Fusion Model's Statistics



(a) Accuracy

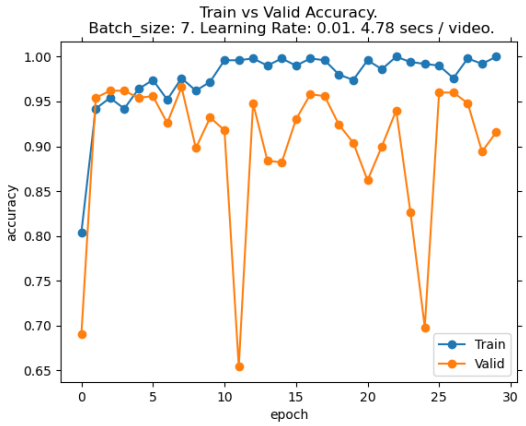


(b) Loss

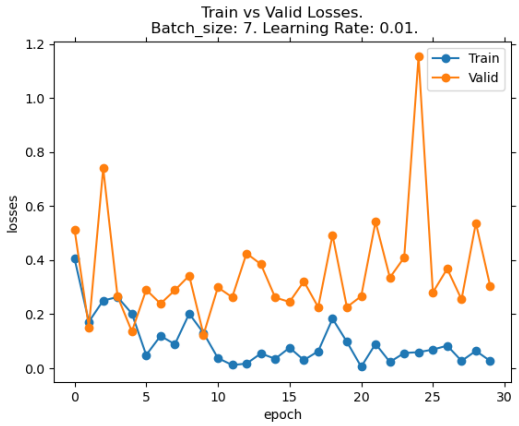


(c) Confusion Matrix

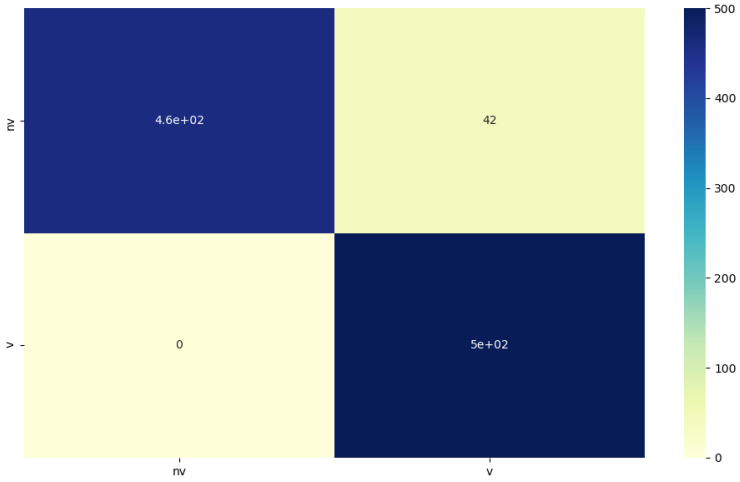
Figure 37: Late Fusion Model's Statistics



(a) Accuracy



(b) Loss



(c) Confusion Matrix

Figure 38: Late Fusion Model's Statistics