UPPSALA
UNIVERSITET

Computer Assisted Image Analysis II (1TD398)

Project Report

Spatio-Temporal Attention for Video Action Recognition.

Jonas Forsman — Vandita Singh

08 March 2020

Contents

1 Introduction **3**
- 1 Motivation . . . 3
- 2 Problem Statement . . . 3
- 3 Limitations and Scope . . . 4
- 4 Report Outline . . . 4

2 Literature Review **5**
- 1 Explainable AI Interpreting, Explaining and Visualizing Deep Learning . . . 5
- 2 Interpreting AI . . . 5
- 3 Explaining and Visualizing AI Systems . . . 6
  - 3.1 Layerwise-Relevance Propagation(LRP) . . . 6
  - 3.2 Video classification with Convolution Neural Networks 8
  - 3.3 Explainable Artificial Intelligence in Action Recognition 8

3 Methodology **9**
- 1 Initial Project Plan . . . 9
- 2 Method . . . 9

4 Implementation **11**
- 1 Data Source . . . 11
- 2 Implementation . . . 11
  - 2.1 Environment . . . 11
  - 2.2 Preparing the data . . . 12
  - 2.3 Action prediction with a CNN-model . . . 13
  - 2.4 Applying Relevance-score and analyzing . . . 15

5 Results **17**
- 1 CNN-model . . . 17
- 2 Analysing one datapoint . . . 17

6 Discussion **21**
1 Discussion and Analysis of Results 21
2 Future Work 28
3 Conclusion 29

Chapter 1

Introduction

1 Motivation

Machine Learning based AI systems are mostly useful for applications for detection of objects in images, understanding of natural languages and processing of speech signals.

These machine learning based Deep Learning models are treated as Black Boxes and are able to reach impressive prediction accuracy, following a nested and non-linear structure which is highly non-transparent[7]. However the impossibility to understand and validate the decision process of an AI system can be seen as a major drawback for these systems. There is no clarity about what information in the input data, made them arrive at the decisions. In case of safety critical systems, reliance of the model on the right features must be guaranteed.

Therefore, the use of explainable and human interpret able AI models is a prerequisite. This can be interpreted as the ability to explain the rationale behind the decisions made, which is also often a crucial aspect in educational context, where students aim to understand the reasoning of their teachers.

The need for Explainable AI is to build systems which are transparent and trustworthy.

2 Problem Statement

The following project has been taken up - "Project 6: Spatio-temporal Attention for Video Action Recognition." With the focus on spatial and temporal attention-based models [3], we want to explore the highly interesting technique of Layer-Wise Relevance Propagation(LRP)[7] that hopefully can help us understand the *black box* of AI (Deep Neural Networks). We do this by

trying to visualize where in an image or video the Deep Neural Network is focusing and thus help to explain the prediction. Presentation for manual inspection is a hotspot overlay on the frames of a video sequence containing the action of interest.

3 Limitations and Scope

Taking into account , the computational complexity, the size of the Dataset was sliced down considerably. Due to limitations in terms of Computing Power , certain methods previously proposed such as Slow Fusion Method for Action Recognition [2] could not be implemented. Evaluation of the quality of Explanations is mostly desirable , however due to time constraints the perturbation analysis (popular measure for measuring heatmap quality) could not be implemented. A few more aspects remain to be present as open challenges in this field.Explainability to the extent where we are only able to view the first order information rather than getting to know the relation between different objects identified as hotspots.Explanations provided by such systems might sometimes be difficult to comprehend and lead to an erroneous analysis.Hence Meta-Explanations remains to be an open area for research in this domain.

4 Report Outline

The report is organized as follows. Chapter II is about background literature followed for the project and the related implementation models have also been discussed.Chapter III presents the model on which the implementation of the system was based.In Chapter IV , the details of the data set HMDB-51 have been presented , with the implementation details. Chapter V comprises of Results obtained from the implementation.Chapter VI focuses on the discussion and conclusion drawn from the proposed task.

Chapter 2

Literature Review

1 Explainable AI Interpreting, Explaining and Visualizing Deep Learning

The simple Machine Learning Systems, based on shallow decision trees and sparse linear models,might be easier to investigate and interpret. When a complex non-linear model for machine Learning (such as a Deep Neural Network with numerous hidden layers) is employed, it gets difficult to interpret the system.Moreover, the predictions made by these models can not be trusted by humans directly due to a lack of having a deeper insight into the system.

The state of art research[4] is going on to build techniques :
(i) Help humans verify whether the black box system is based on correct input variables to make predictions.
(ii) Explain-ability of the decisions of Artificial Intelligence System
The explanability in terms of the features on which the predictions are based need to be verified by the humans.This further helps to build trustworthy systems.

2 Interpreting AI

This aspect focuses on verifying whether the concepts that the deep learning model has learned based on the correct input variables and if the model combines them in a meaningful manner.[4].

Following are some methods focusing on current research to bring more interpret-ability into complex Deep Learning Systems. Nyugen et. al [14] is based on Activation Maximization where the proposed solution builds a

prototype in input domain that maximally activates a certain output neuron of the such that the prototype looks realistic.[14]. Hong et. al [10] propose an interpretable text-to-image synthesizer in which th proposed layout generator progressively constructs a semantic layout in a coarse to fine manner[10].This allows the model to generate semantically more meaningful images that are as realistic as possible and easy for the humans to interpret.Hu et. al. [11] emphasize on bringing useful invariances into the model which is the key to the interpret-ability of a model.The authors present a learning algorithm for summarization of data into clusters.Through a self-augmented training strategy , the cluster structure is driven to be human interpretable. Most of the above mentioned techniques keep the goal to better understand the input data.

3 Explaining and Visualizing AI Systems

To identify whether the input features that have contributed to the given prediction or if these features are locally prevalent for the prediction is a major concern. Having large datasets from heterogeneous sources may pose a high risk of picking the wrong fetaures to support the decisions and become "Clever Hans" predictors.[12]

Fong and Vedaldi [8] present a technique called "Meningful Perturbation" which synthesizes a minimal local perturbation of the current data point that maximally affects the ML decision. [8]. In this approach the relevance for the decision is understood if the variables form the peturbation. Synthesis can be computationally expensive while high flexibility is conferred in shaping the perturbation. Ancona et.al. focus on Gradient Based Techniques where the explanation is derived from extracting the components of the gradient of the decision function.[6]. This method is based on calculating automatic differentiation mechanism available in most neural network libraries. However these could be computationally expensive.

3.1 Layerwise-Relevance Propagation(LRP)

Bach et. al [1] suggests techniques that relate to other classes of explanation techniques based on propagation. One such method is the Layer-wise Relevance Propagation. Monatven et. al [13] present the "Layerwise Relevance Propagation" model for explainable AI. The Layerwise-Relevance Propagation (LRP) is a technique that aims to provide explainability for the decisions made by deep learning system. This model operates by propagating the prediction backwards in the neural network, using a set of purposely defined

local propagation rules. The propagation procedure implemented by LRP is subject to a conservation property, where what has been received by a neuron must be redistributed to the lower layer in equal amount. [13]. The general framework for LRP procedure has been shown in Figure 2.1

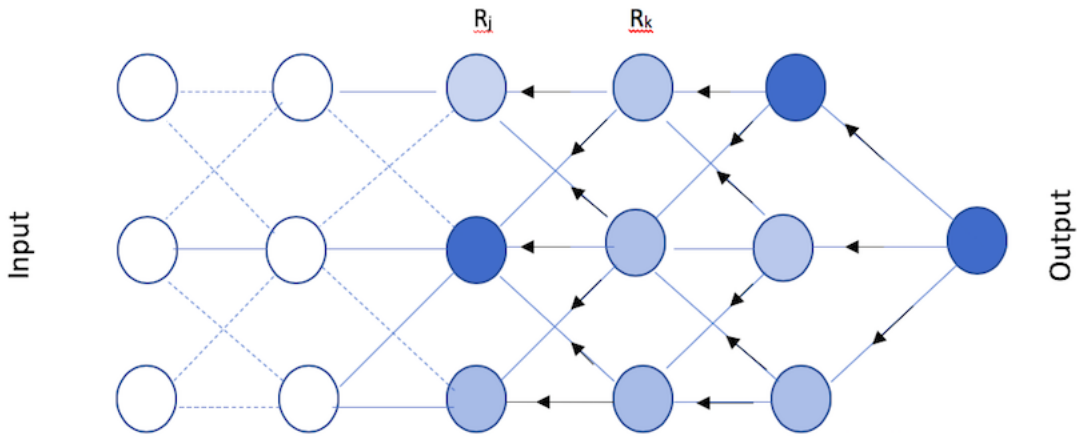


Figure 2.1: Layer-wise Relevance Propagation: Framework

Let j and k be two neurons at two consecutive layers of the neural network. Propagating relevance scores (R_k), at a given layer on to neurons of the lower layer is achieved by applying the rule [13]:

$$\mathrm{R}_j = \Sigma_k \frac{z_{jk}}{\Sigma_j z_{jk}} Rk$$

In the Propagation Rule ,(i)to model the quantity to which the neuron j has contributed to make the neuron k relevant, an important role is played by the quantity : z_{jk}

(ii) To enforce the conservation property , the denominator plays a key role:

$$\Sigma_j z_{jk} Rk$$

Once the input features have been reached, the propagation procedure terminates. If the above rule is applied to all the neurons in the network, the layer-wise conservation property can be easily verified i.e.

$\Sigma_j R_j = \Sigma_k R_k$

Also, the global property,

$\Sigma_i R_i = f(x)$

would be conserved, where f(x) is the prediction made by the neural network.

3.2 Video classification with Convolution Neural Networks

Karpathy et.al.[2] propose an extensive empirical evaluation of Convolution Neural Networks (CNNS) on large scale video classification. Multiple Approaches-Single Frame, Late Fusion, Early Fusion and Slow Fusion were discussed for extending the approach in time domain taking into consideration the local spatio-temporal information and speed up the training process.

3.3 Explainable Artificial Intelligence in Action Recognition

Samek et. al [7]present Sensitivity Analysis and Layerwise-Relevnce Propagation for explainability of AI for Image classification , text document classification and human action recognition. The Human Action recognition was based on SVM Classifier. HMDB51 dataset[9] was used for performing the evaluation.The results show the video frames for action sit-up where the authors found that the model focuses on the upper body of the person and thus made good sense.An important finding suggested by the author is that relevance score not only visualizes the relevant locations of the action within a video frame but also identifies the most relevant time points within the video sequence[7].

Meng et. al [3] propose a method for spatio-temporal explainabilty for Video-Action Recognition. For spatial attention, the authors employ a saliency mask to allow the model to focus on the most salient parts of the feature maps.[3] For temporal attention, a convolutional LSTM based attention mechanism was implemented to identify the most relevant frames from an input video.[3]

Chapter 3

Methodology

1 Initial Project Plan

- Get data HMDB51, unpack and investigate
- Choose four different categories (hug, punch, throw, fall) , later four more categories were added (run,smoke, wave and jump);Total eight categories
- Preparation of the dataset with desired frame selection and adequate pre-processing
- Perform Action Prediction:
  Step 1:Train a Deep Neural Network model M with dataset , that can classify the 8 categories
  Step 2:Testing
- Implementation of Layer-wise Relevance Propagation
- Analyse the outcome

2 Method

The work flow for the project can be seen in Figure 3.1

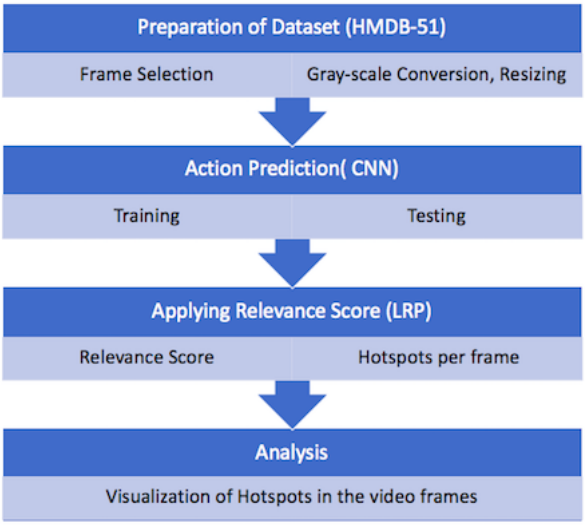


Figure 3.1: Workflow

Deep Learning System for Action Recognition (HMDB-51) For our project we needed a CNN that was trained in a way that incorporated both the spatial and the temporal data. Our first approach was to use an CNN architecture proposed by A.Karpathy et. al [2]. More precisely we wanted to implement the *Slow Fusion* architecture. It soon turned out that this architecture would produce a CNN that was way to big and demanding for the hardware we had available in the project. We had to settle for a much simpler solution. The simpler architecture consists of two convolution layers, one max pooling layer, one flatten layer and two dense layers. When used with 100x100 frame size it will have more than 75 million parameters.

Chapter 4

Implementation

1 Data Source

The dataset we have choosen is HMDB51[9]. It consists of action annotated videos. We selected a subset of the actions [fall, hug, jump, punch, run, smoke, throw, wave]. The videos are in avi format.

Fall 136 files 29MB
Hug 118 files 36MB
Jump 151 files 30MB
Punch 126 files 31MB
Run 232 files 60MB
Smoke 109 files 41MB
Throw 102 files 32MB
Wave 104 files 28MB

TOTAL 1180 files 319MB

2 Implementation

2.1 Environment

This project was implemented in several steps with different development environments. In the first stage we used MathLab. For the second stage we used a Virtual Machine with the following installed:
- Jupyter Notebook
- Python

- Keras
- TensorFlow
- iNNvestigate

2.2 Preparing the data

When preparing the data we had to consider that we needed many datapoints to get as good CNN when training as possible, but that each datapoint should not be to big due to hardware limitations. Preparing the data was done in several steps. We started with downloading the dataset and the extracting the eight categories of interest. Each category had a folder containing short videos with the annotated action. From this we wanted to create datapoints that would be usefull for training a CNN-model. The datapoints was prepared in the following steps using MATLAB:

Datapoint identity Each datapoint created from video was named as follows, *vcCATEGORYCODEnnnn_ff*.

The category code is one of the following:

F - fall
H - hug
J - jump
P - punch
R - run
S - smoke
T - throw
W - wave

The nnnn is a number between 1000-9999 and the ff is the first frame offset.
Example: BATMAN_BEGINS_fall_floor_f_cm_np1_ba_med_11 was named to vcF1012_2.

Frame selection We wanted each datapoint to consist of 16 frames from the video. Depending on the length of the video we extracted every 3rd, 4th, 5th or 6th frame. The frames where stored in a folder named with the

datapoint identity.

Color or gray scale We wanted to reduce the amount of data used when training the CNN-model so we decided to convert all frames from RGB to gray scale images.

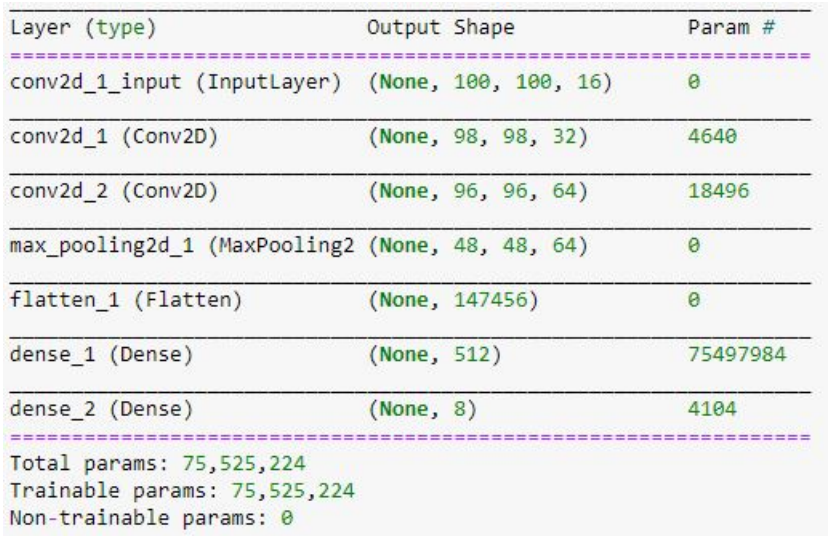
Resizing Another decision to reduce the datapoint size was to resize the frames. We tried 50x50 and 100x100, producing two versions of the datapoints.

2.3 Action prediction with a CNN-model

The development of the CNN-model was done with Keras and a TensorFlow backend, using Python and Jupyter Notebook as frontend.

CNN architecture The Convolution Neural Network proposed to be implemented can be seen in Figure 4.1.

Datasets We divided the datapoints in three different datasets. The training dataset containing 1687 datapoints. The validation dataset containing 382 datapoints and the test dataset containing 363 datapoints. All datasets where randomly selected to create an even distibution. We randomized based on the *original video* to avoid very similar datapoints appearing in more then one dataset, meaning that two datapoints that originate from the same video do not end up in different datasets.



Layer (type)	Output Shape	Param #
conv2d_1_input (InputLayer)	(None, 100, 100, 16)	0
conv2d_1 (Conv2D)	(None, 98, 98, 32)	4640
conv2d_2 (Conv2D)	(None, 96, 96, 64)	18496
max_pooling2d_1 (MaxPooling2	(None, 48, 48, 64)	0
flatten_1 (Flatten)	(None, 147456)	0
dense_1 (Dense)	(None, 512)	75497984
dense_2 (Dense)	(None, 8)	4104

Total params: 75,525,224
Trainable params: 75,525,224
Non-trainable params: 0

Figure 4.1: CNN model summary: 100x100 pixel frames

Training of model When we train our CNN we first tried with the dataset based on frame size 50x50 and 16 frames. The training was fairly quick, it took about 15 minutes (09:46:35 - 10:02:38) and got an accuracy scores of 0.4918032757571486 with a loss of 4.338693011653879.
We then proceeded to the other dataset based on frame size 100x100 and 16 frames. Now the training took longer time, about 39 minutes (17:44:23 - 18:23:28). This time we got an accuracy score of 0.47643978605095627 with a loss of 5.188817363758986. We used epochs = 20 and batch size = 100. Trying different numbers for epochs and batch size didn't improve the result.

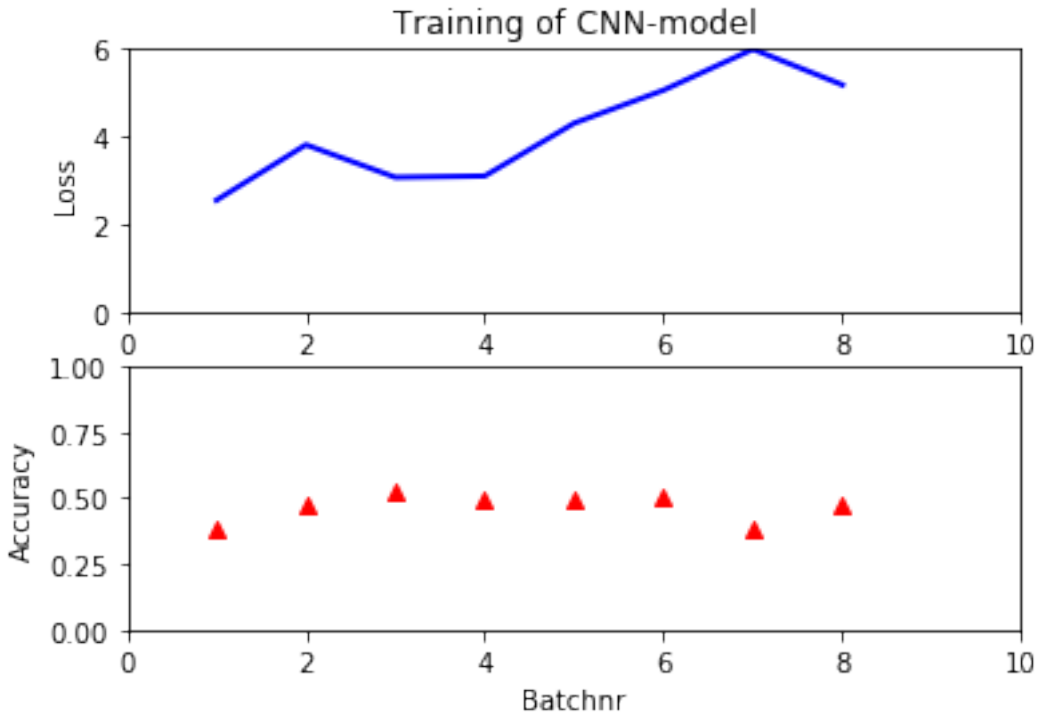


Figure 4.2: CNN model training process

Verifying the model We decided to use the model based on 100x100 frame size even if it didn't reach higher. This was mostly due to the fact that later on when we try to "explain the AI" it's important for the human to actually see what's in the frame and that turned out to be hard when using 50x50 frames. The test dataset used to verify the model contains 363 datapoints and it resultet in an accuracy of 0.47107438016528924. Only slightly worse than when training the model.

2.4 Applying Relevance-score and analyzing

We decided to use a framework, iNNvestigate [5] to implement Layer-wise Relevance Propagation.

Preparing the model To use the LRP on a model you first have to prepare it by making a copy and removing the final softmax layer. Then you can use that model for retreiving an analyzer object that will give you the relevance score for each datapoint provided to it.

Relevance score The relevance score is a number per input feature i.e pixel in this case, that tells us something about how relevant the input was to the classification made. The relevance score can be negative, meaning that this particular input spoke against the classification made.

Hotspots Focusing on input data with high relevance score, a.k.a hotspots could give us an idea what is important for the classification. In our case we decided to set a threshold at 0.3 and creating an hotspot overlay for each frame for manual analys.

Analyzing We already have a hotspot overlay function and a Relevance Heatmap for analysing the model, but in order to investigate how the frames in the video sequence differ in relevance, we created two visualization functions. One that shows the sum of the relevance per frame and one that shows the sum of hotspots per frame.

Chapter 5

Results

1 CNN-model

Testresult from CNN-model Applying the testdataset to the model gave the following result. Correct classification on 171 datapoints out of 363 (48%). Wrong classification on 192 datapoint out of 363 (52%).

Distribution of categorization by CNN-model The distribution is not even. Some categories stand out more like *smoke* and *fall* which did get a very low correct classification.
F - fall: 1 corr (5%), 20 fail
H - hug: 39 corr (56%), 30 fail
J - jump: 14 corr (36%), 24 fail
P - punch: 51 corr (74%), 18 fail
R - run: 30 corr (43%), 39 fail
S - smoke: 0 corr (0%), 20 fail
T - throw: 12 corr (60%), 8 fail
W - wave: 24 corr (42%), 33 fail

2 Analysing one datapoint

Analyzing datapoint vcP1269_3 we first have a look at the 16 frames that make up the datapoint. Here (figure 5.1) with a hotspot overlay.

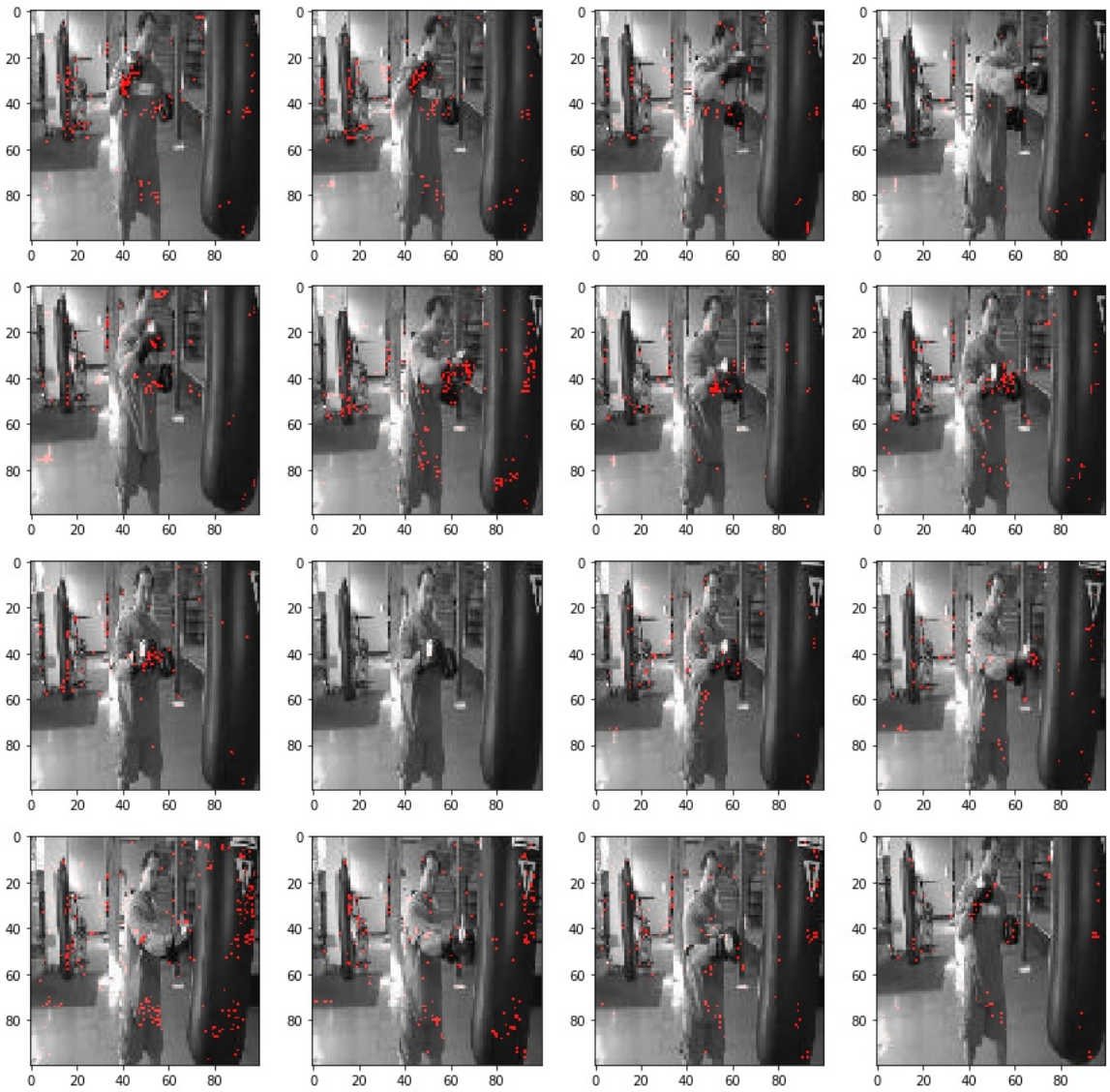


Figure 5.1: Datapoint vcP1269_3 all frames with hotspot overlay

Heatmap and hotspot overlay Focusing on the first frame, the heatmap and hotspot overlay can give a good understanding of what is relevant to the CNN model at this instant, as can be seen in figure 5.2.

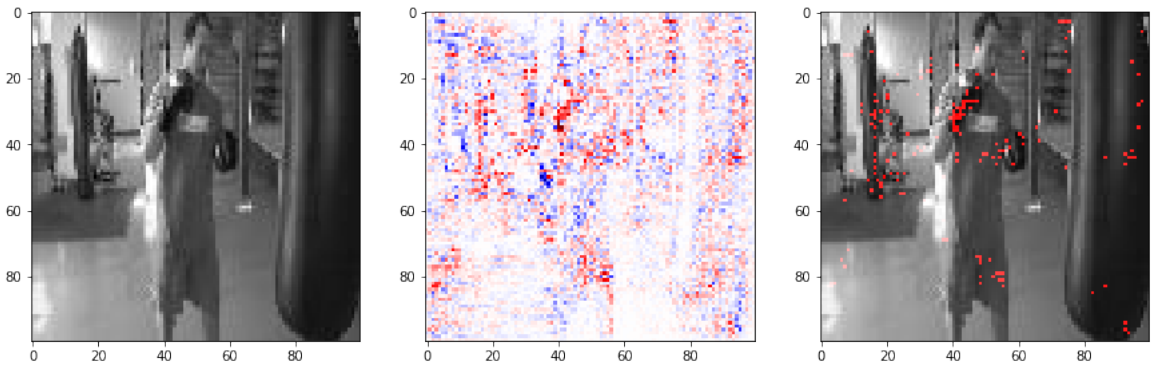


Figure 5.2: Datapoint vcP1269_3 analyzed with heatmap and hotspot overlay

Relevance and hotspot distribution The relevance and hotspots are not evenly distributed among the frames in a video sequence, and we wanted to manually investigate this looking at the relevance distribution in figure 5.3 and the hotspot distribution in figure 5.4. Interestingly frame 2, 6 and 13 all have a high amount of relevance indicating that they are all important but frame 2 have way more hotspots. Frame 6 and 13 are also very similar. Frame 5 also has alot of hotspots but it also have alot of negative relevance making it the odd frame in the collection.

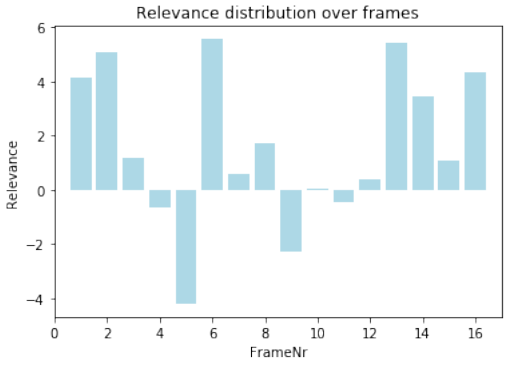


Figure 5.3: Datapoint vcP1269_3 relevance distribution

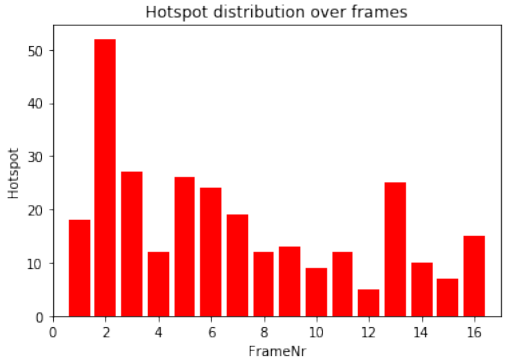


Figure 5.4: Datapoint vcP1269_3 hotspot distribution

Chapter 6

Discussion

1 Discussion and Analysis of Results

In the images for Relevance Scores, the Relevance Score is present in form of Blue and Red dots. The Red dots depict a positive contribution to the prediction, a Blue dot depicting negative contribution while the white ones depicting a neutral zones.(Figure 6.1)

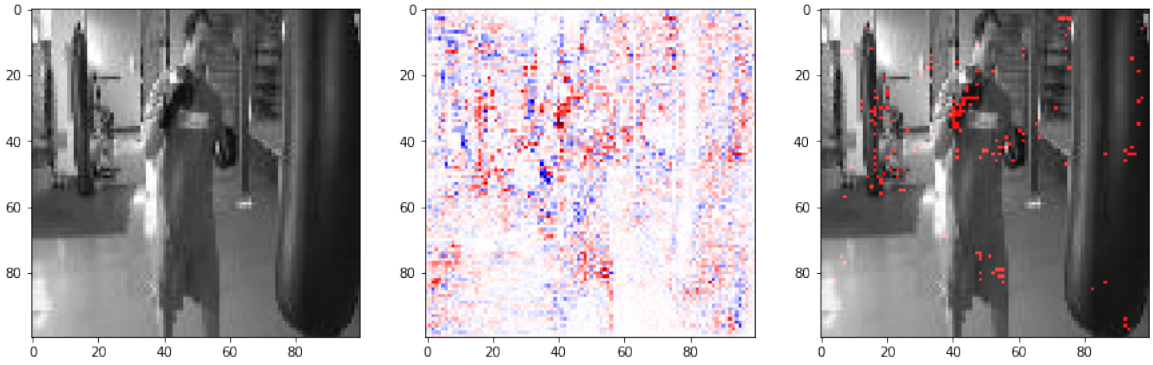


Figure 6.1: Action:Punch with Relevance Score and Hotspots(L-R)

In Figure 6.2 and Figure 6.5 the hot spots continue to be over the person performing the jump (seen as clusters of Red dots) in the frames.

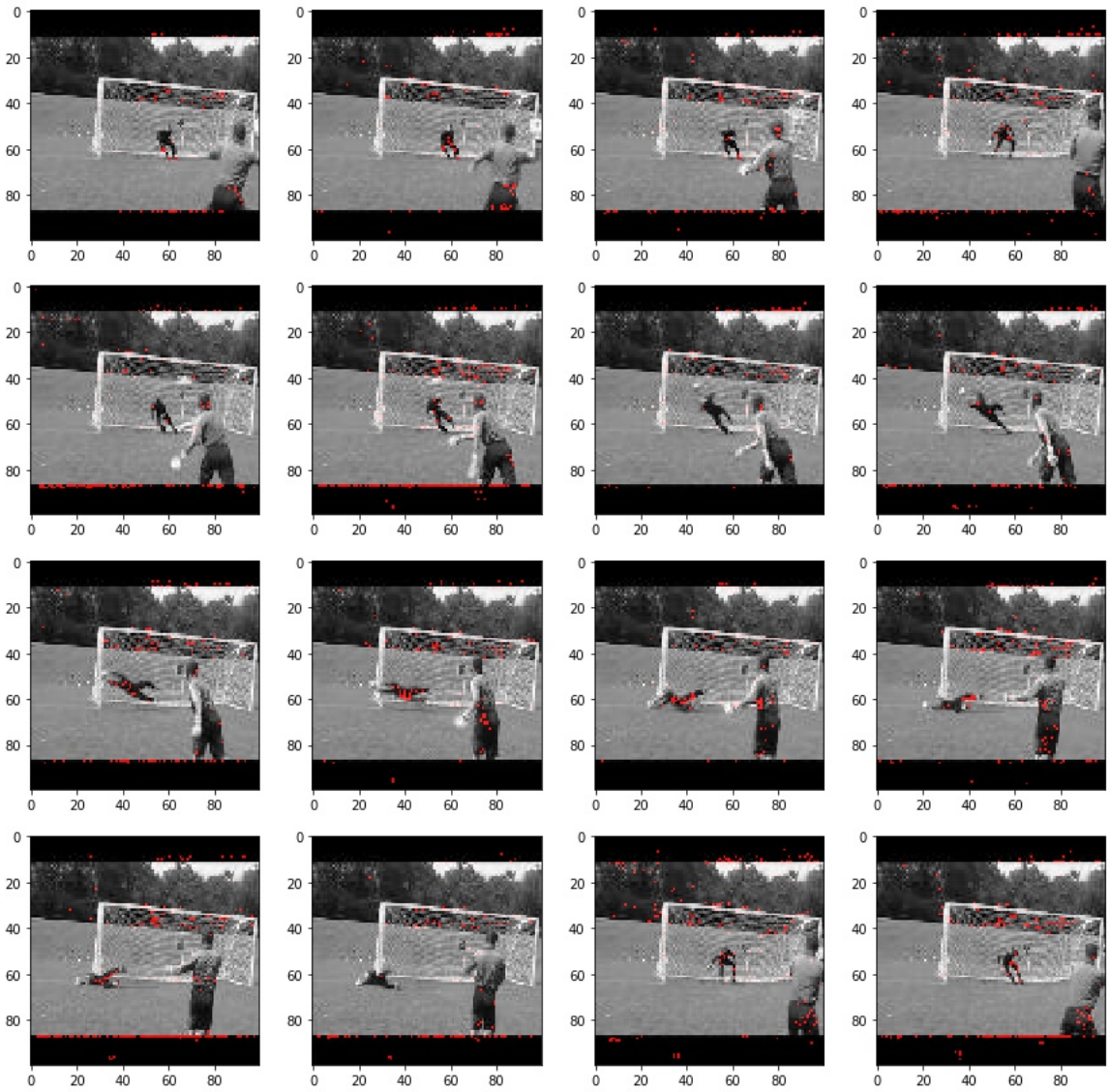


Figure 6.2: Action:Jump Video 1 Frame-wise Hotspots

The problem arises when the hotspots can be seen in the lower part of the frames which clearly seem to be present out of the scene. Also, certain objects in the background , such as the trees (Figure 6.2 and Figure 6.5 also are marked as hot spots, which clearly questions the explainability of the AI system.

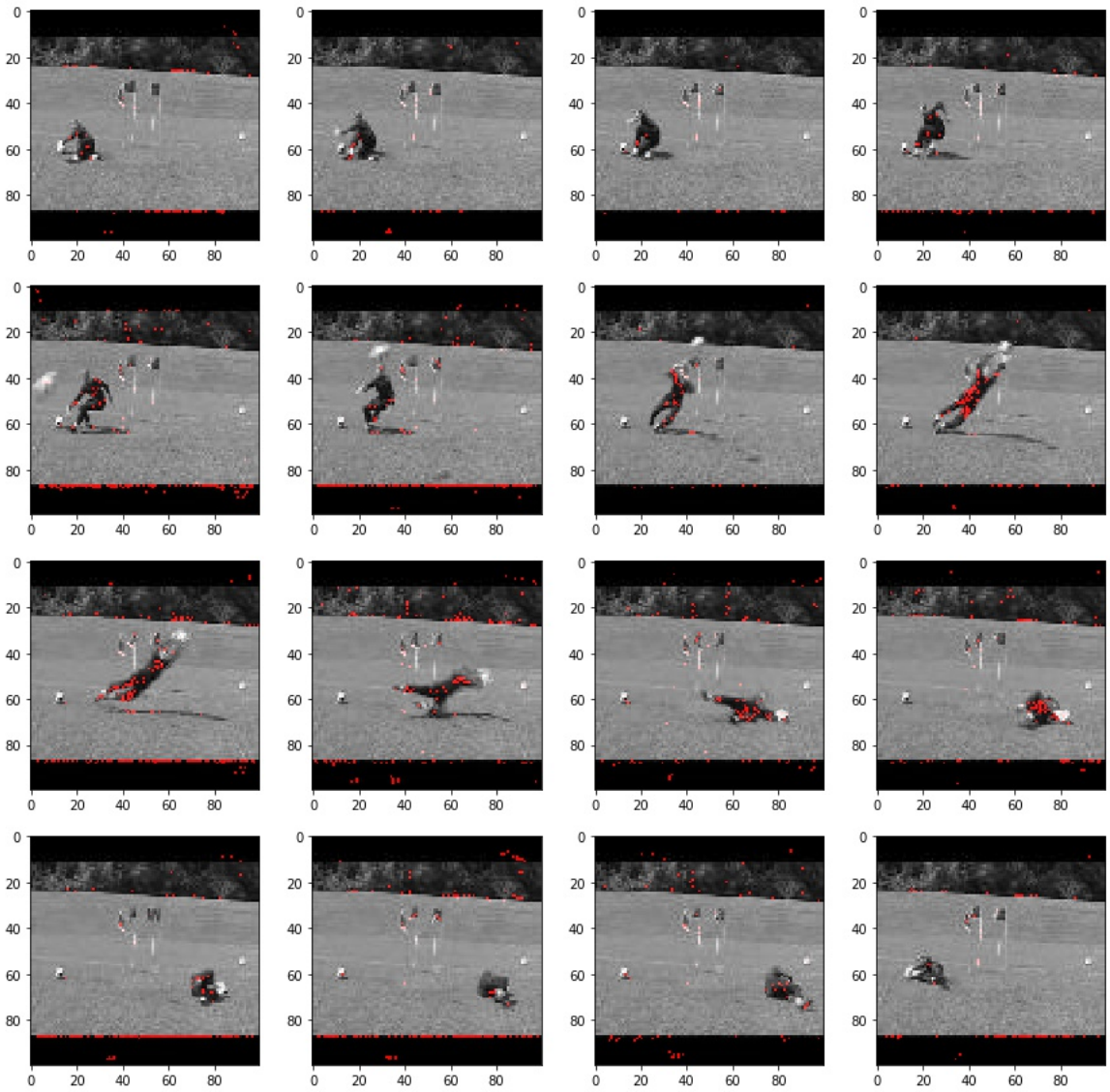


Figure 6.3: Action:Jump Video 2 Frame-wise Hotspots

The clusters of Red Dots in Figure 6.4 appear over the gloves of the person to predict the action "Punch".

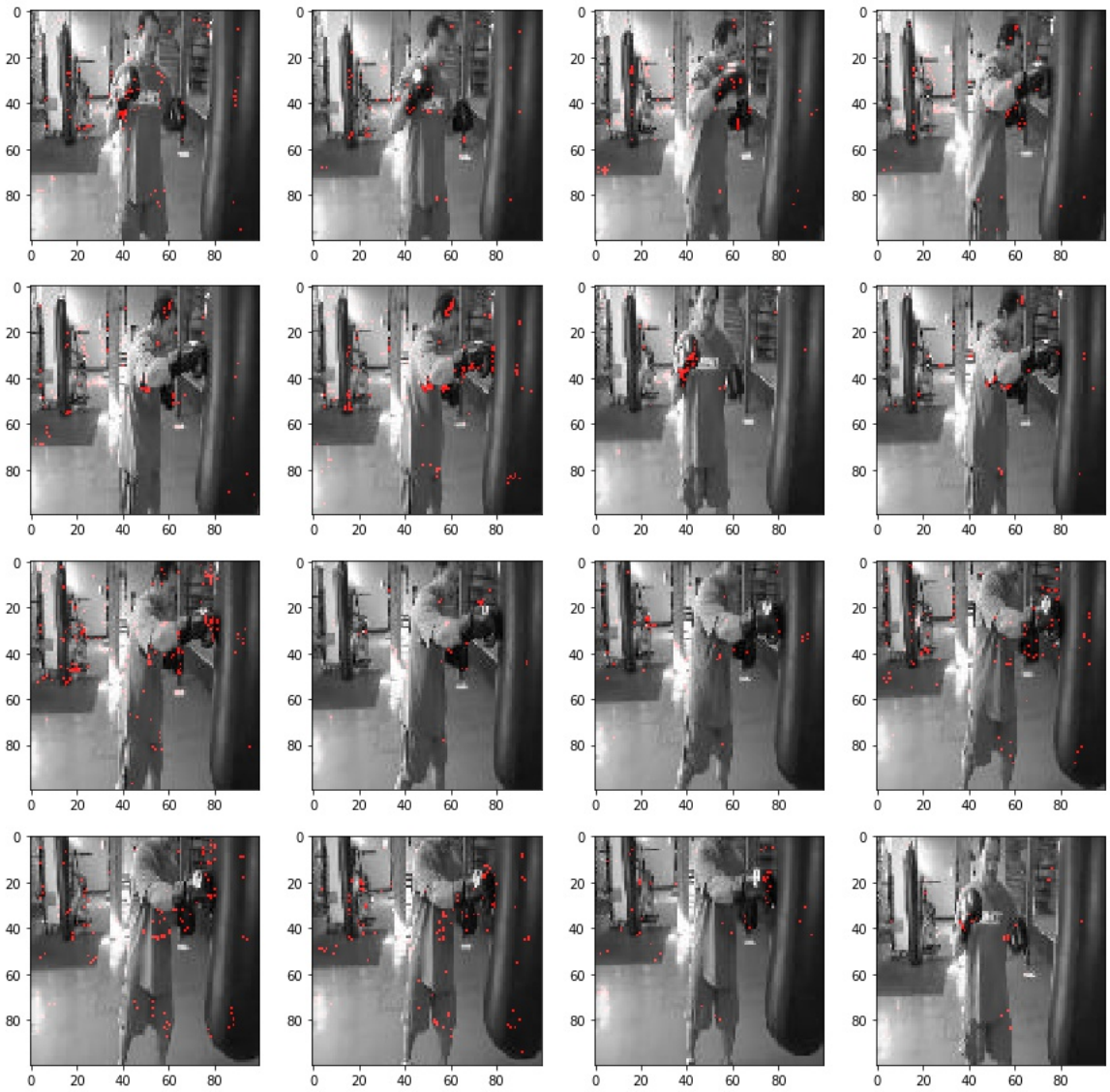


Figure 6.4: Action:Punch Video 1 Frame-wise Hotspots

However , there can be seen some focus on the punching bag in the background of the image. This poses a question whether the computation of Relevance Score really reliable for explain-ability of Deep Learning methods as it is evident of taking the background information into consideration.

In Figure 6.5 , the hot spots can be seen to appear on the punching bag which might be due to the reason that the action "Punch" was predicted in context with the presence of the punching bag. This may not always be the case. So , this definitely puts the explainablity of the AI system under question.

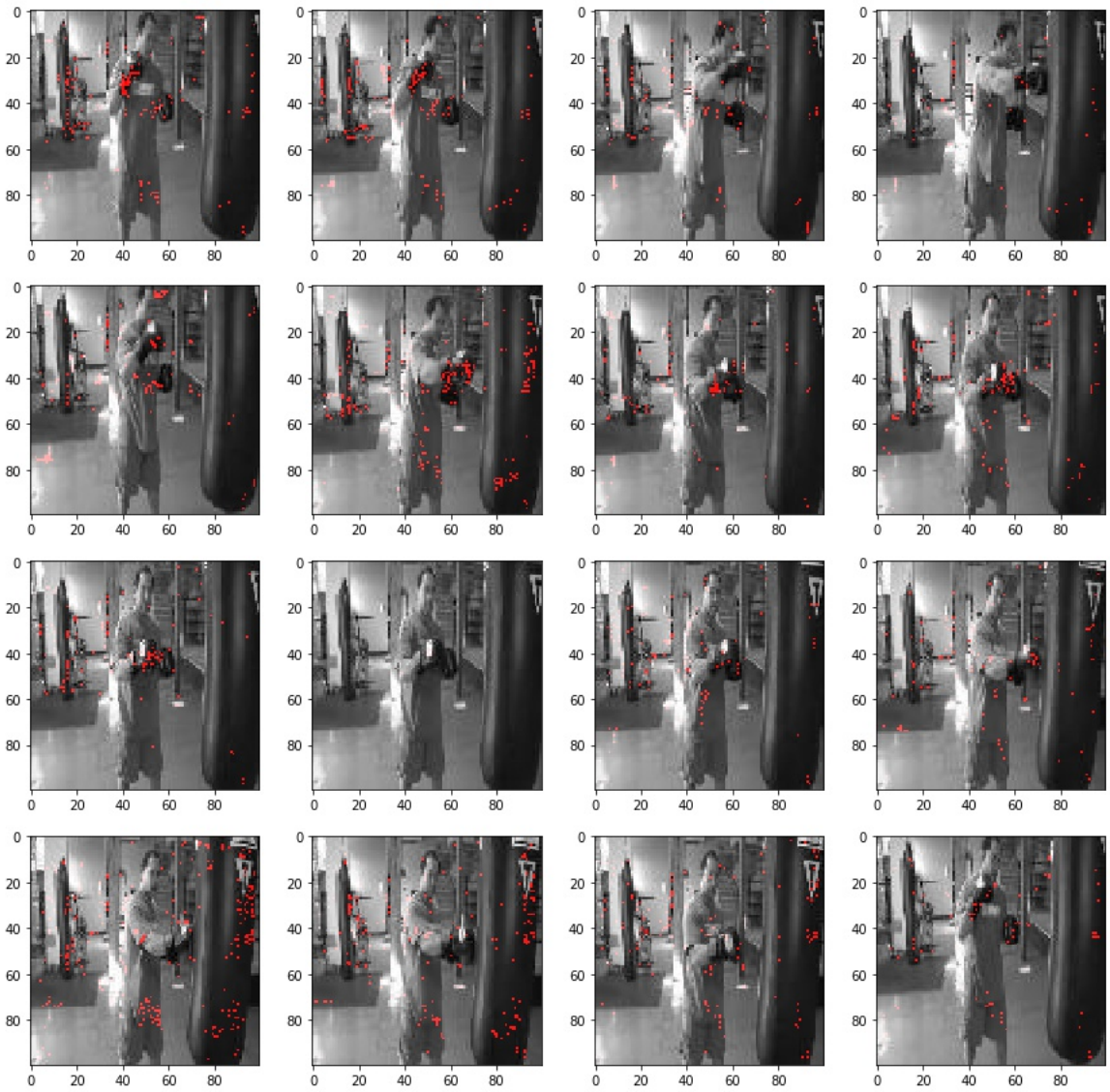


Figure 6.5: Action:Punch Video 2 Frame-wise Hotspots

In Figure 6.6, the hotspots do appear on the arms of the men fighting, but the frames 1-9 and 14 the hotspots can also be seen on the window in the background(but the darker ones only).This indicates the possibility of the Neural network to be biased towards the darker areas in any frame.

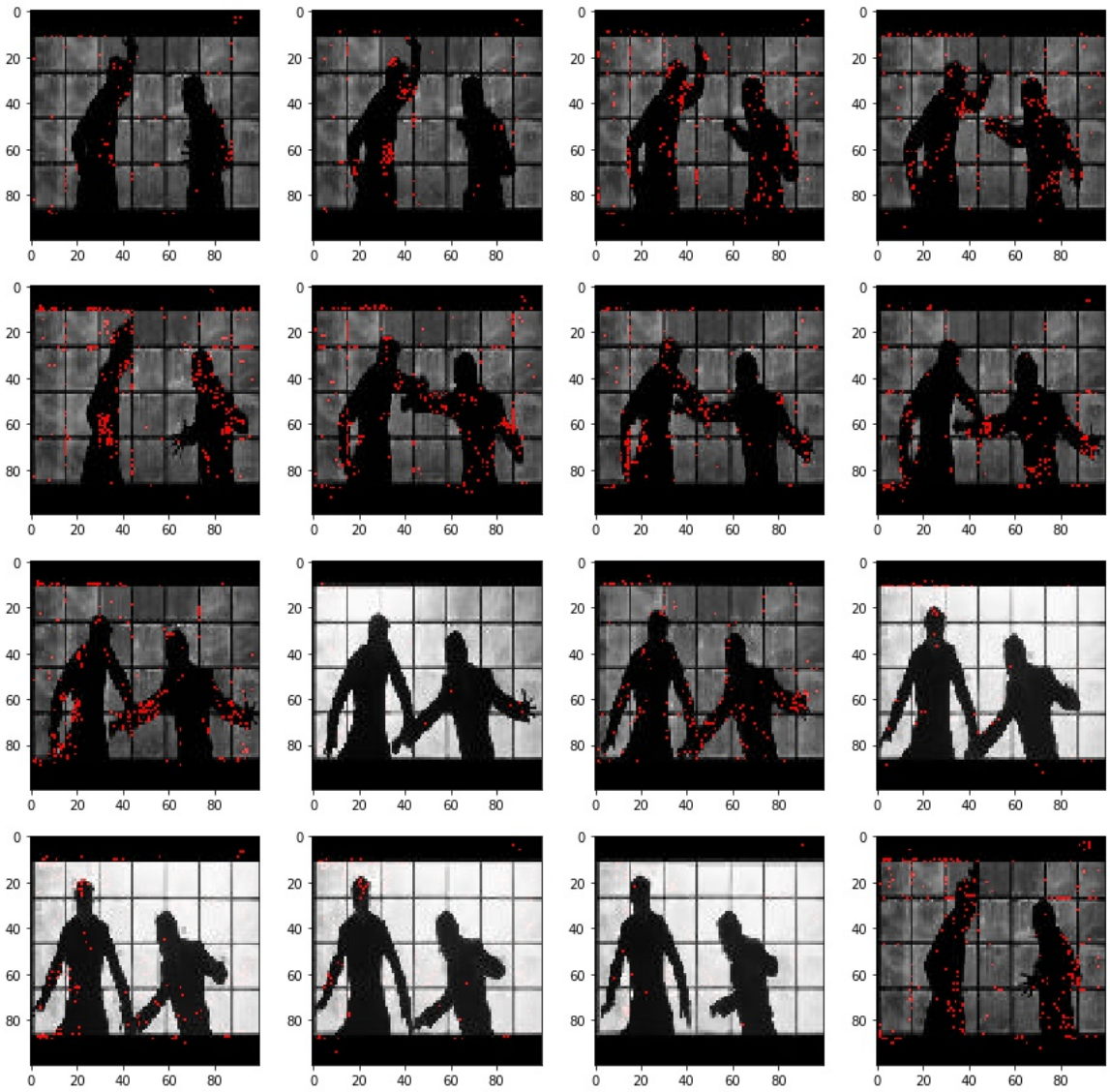


Figure 6.6: Action:Punch Video 3 Frame-wise Hotspots

The Figure 6.7, the hotspots do appear on the legs of the person running in all the frames, however Frames 1- 6 focus on the car present in the background, Frame 7 on the road in the background and then in frames 8-16 , the focus appears to be present on the tree in the background.This could pose a possibility that the gradual change of objects in the background be possibly contributing in predicting an action involving motion. If this holds to be true, the predictions would fail for motion based actions in videos where the background objects remain almost similar, such as forests/highway.

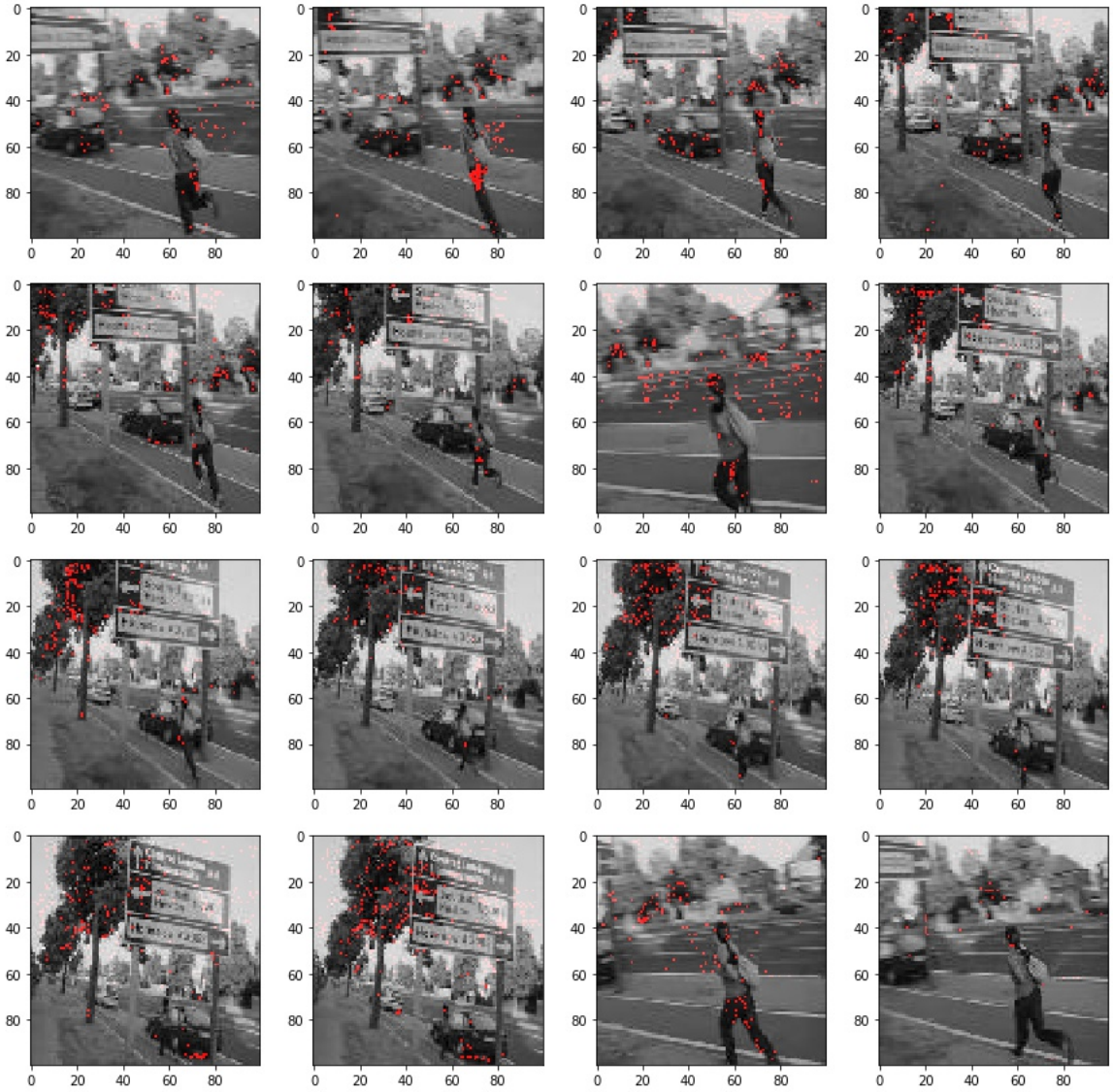


Figure 6.7: Action:Running Frame-wise Hotspots

The Figure 6.8 shows the frames for a video where the predicted action was "Wave".This is a scene from a movie where different persons are involved. Frames 1 to 6 ,8 to 14 have the image of the person who is waving his hand but the remaining frames have some other person's image. If the scene is switched as in this case , then no hot spots should have been spotted in scenes where no waving is being performed while, in this case such frames with almost no useful information can also be seen to be having a good number of hot spots.This could possibly lead to a false prediction and does not stand out to be a robust technique for explainabilty of AI systems.

Figure 6.8: Action:Waive Frame-wise Hotspots

2 Future Work

The work could be extended to extraction of hot spots on the segmented objects in the image. In this case the background can be separated from the video frames and then the observation be made as to what points of focus we get in form of Relevance Score.

Evaluation of the Explanation may be performed by means of Perturbation Analysis [4] to identify if the relevance score is actually related to

relevant input data or not.

Implementation of this system by incorporating Slow Fusion process for spatio-temporal Action recognition in Videos[2].

These results could be compared with the LSTM implementation for the same dataset.

3 Conclusion

The aim to visualize the deep learning system for spatio-temporal Action Recognition was fairly achieved in terms of interpret ability where the results from Layerwise-Relevance Propagation system produce images with Relevance Scores (Red Dot clusters) of the pixels in the frame. The focus was found mostly to be on objects, backgrounds and objects of contextual importance (such as punching bag in case of "Punch"). There seems to have been some bias towards the darker intensities in the frames. The result when it comes to *Explainable AI* was inconclusive but that can also be an effect of the low quality of the CNN used.

Bibliography

[1] Bach Sebastian et. al. *On pixel-wise Explaninations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation*. URL: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0130140.

[2] Karpathy Andrej et. al. "Large-scale Video Classification with Convolutional Neural Networks". In: (). URL: http://cs.stanford.edu/people/karpathy/deepvideo.

[3] Meng Lili et. al. "Interpretable Spatio-temporal Attention for Video Action Recognition". In: (2019).

[4] Samek W. et al. *Explainable AI:Interpreting, Explaining and Visualizing Deep Learning*. Springer, 2019.

[5] Maximilian Alber. *iNNvestigate*. URL: https://github.com/albermax/innvestigate.

[6] M. et. al. Ancona. "Gradient Based Attribution Methods." In: *Explainable AI:Interpreting, Explaining and Visualizing Deep Learning, Springer,Cham* 11700 (2019), pp. 169–191.

[7] Samek Wojciech et.al. "Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models". In: (2017). DOI: arXivpreprintarXiv:1708.08296.

[8] A. Fong.R. Vedaldi. "Explanations for attributing deep neural network predictions." In: *Explainable AI:Interpreting, Explaining and Visualizing Deep Learning, Springer,Cham* 11700 (2019), pp. 149–167.

[9] HMDB-51. *A large-Human Motion Database*. URL: http://serre-lab.clps.brown.edu/resource/hmdb-a-large-human-motion-database/,%20Retrieved%20January%2021,%202020.

[10] S. et al. Hong. "Interpret able Text to Image Synthesis with hierarchical semantic layout generation". In: *Explainable AI:Interpreting, Explaining and Visualizing Deep Learning, Springer,Cham* 11700 (2019), pp. 77–95.

[11] W. et al. Hu. “Unsupervised discrete representation learning.” In: *Explainable AI:Interpreting, Explaining and Visualizing Deep Learning, Springer,Cham* 11700 (2019), pp. 97–119.

[12] S. et. al Lapuschkin. “Unmasking Clever hans predictors and assessing wht machines really learn.” In: *Nat. Commun. 10* 1096 (2019).

[13] G. et. al. Montavon. “Layerwise Relevance Propagation: An overview”. In: *Explainable AI:Interpreting, Explaining and Visualizing Deep Learning, Springer,Cham* 11700 (2019), pp. 193–209.

[14] A. et al. Nyugen. “Understanding Neural Networks via feature visualization”. In: *Explainable AI:Interpreting, Explaining and Visualizing Deep Learning, Springer,Cham* 11700 (2019), pp. 55–76.