Experiment report -2

Presented by: Amit Hayun, Bar Loupo
Lecturer: Dr. Marina Litvak

Task definition

Following the conclusion of experiment number-1.

- We will try a different gradient descent optimization algorithm to explore the different performance of the model.[see experiment hyperparameter section]

- In order to thoroughly examine our model performance in a real-time environment, we built a prototype **Abuse Detection System [ADS].**
  We bought an IP camera and built a Data Stream pipeline that takes as an input video frame capture by the IP camera, and as an output returns an np frame of size(149,224,224,5) .
  The output will use as input for the model in order to make a prediction on the frames [see ADS evaluation section]

The **model architecture**, **workspace environment**, **data preprocessing**method, **data augmentation techniques**, and **Data sets**are the same as in experiment number 1.

Experiment hyperparameter

We chose to try the Adam optimization algorithm, with the command value for the parameters $\beta 1$ $\beta 2$ ε as shown**in Table 1**

Table 1:experiment hyperparameter

learning rate α	**0.01**
epsilon ε	**1e-07**
beta $\beta 1$	**0.9**
beta $\beta 2$	**0.999**
batch size	**6**
Number of workers	**6**
Number of epoch	**30**
GPU	**1x Titan RTX**

Result

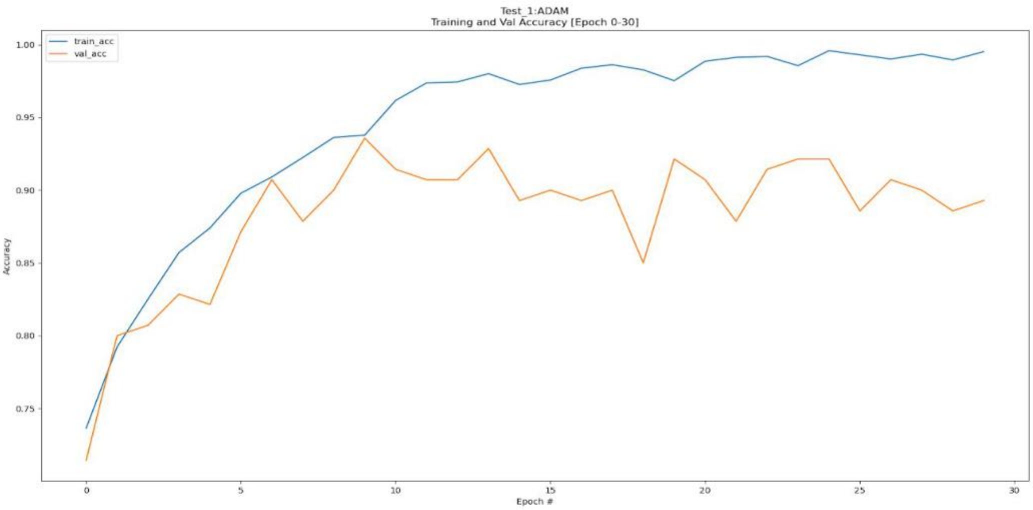
Training result

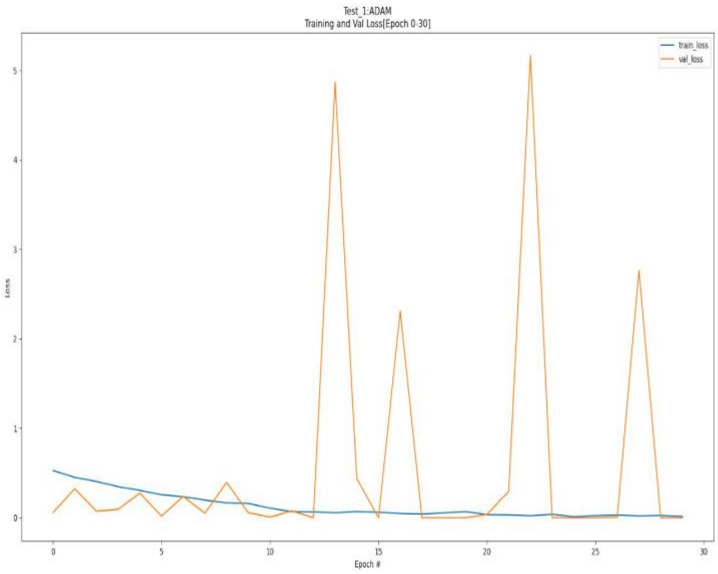
we search for the minimum gap between Test error and Val error

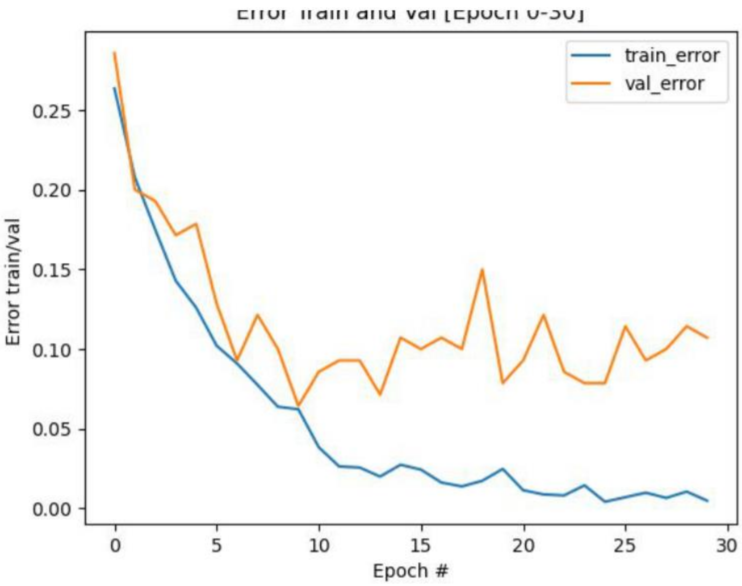
$$Test\ Error = 1 - Test\ Accuracy$$

$$Val\ Error = 1 - Val\ Accuracy$$

The following images show the results of training and Val accuracy, loss, error during the experiment



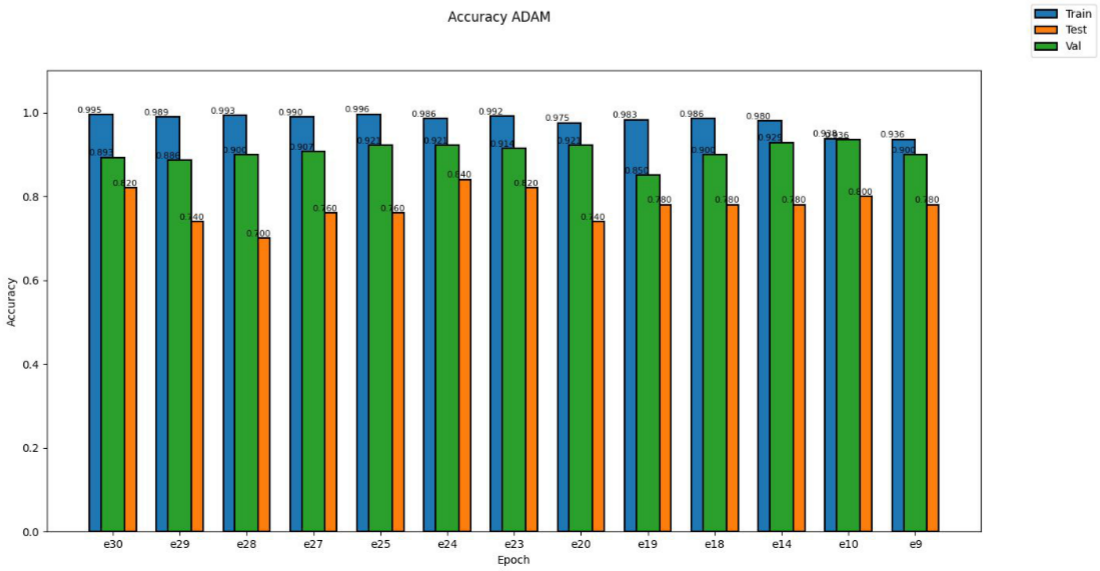


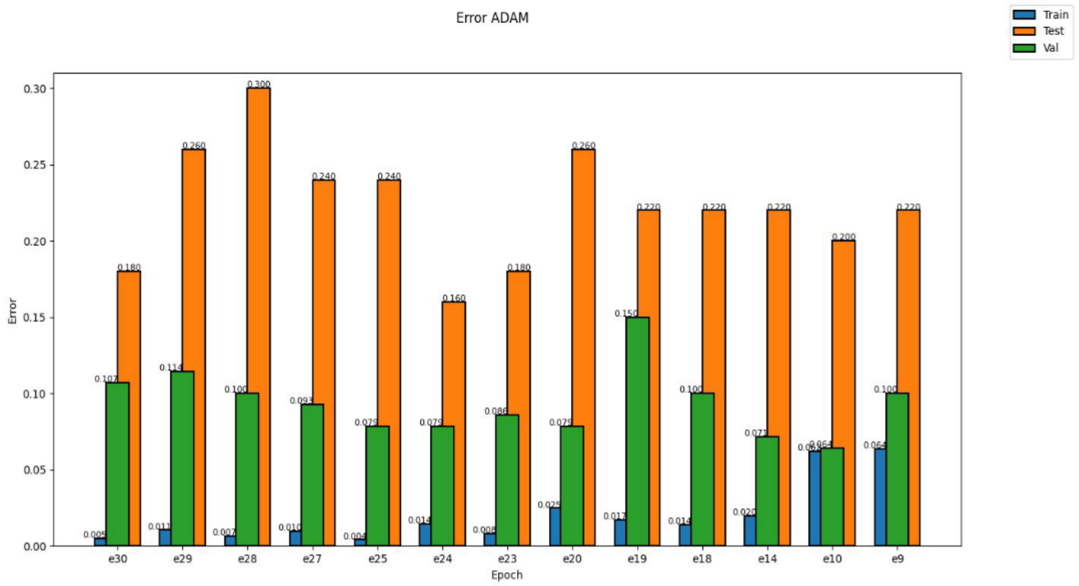


We selected the models with the lowest error rate and highest accuracy on the dev set, and found the following models to be the most promising for testing on our test set: models at epoch number [30,29,28,27,25,24,23,20,19,18,14,10,9].

After testing those models we will be analyzing the model performance on our test set using evaluation metric and then choose the best model by this measurement.

the following image shows the Training and Val result accuracy, error, on the chosen model





Evaluation metrics

We used the same evaluation techniques as stated in the experiment number 1 report.

$$Recall = \frac{TP}{TP+FN}, \quad Precision = \frac{TP}{TP+FP}, \quad Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

$$F1_{score} = 2 \times \frac{Recall \times Precision}{Recall + Precision},$$

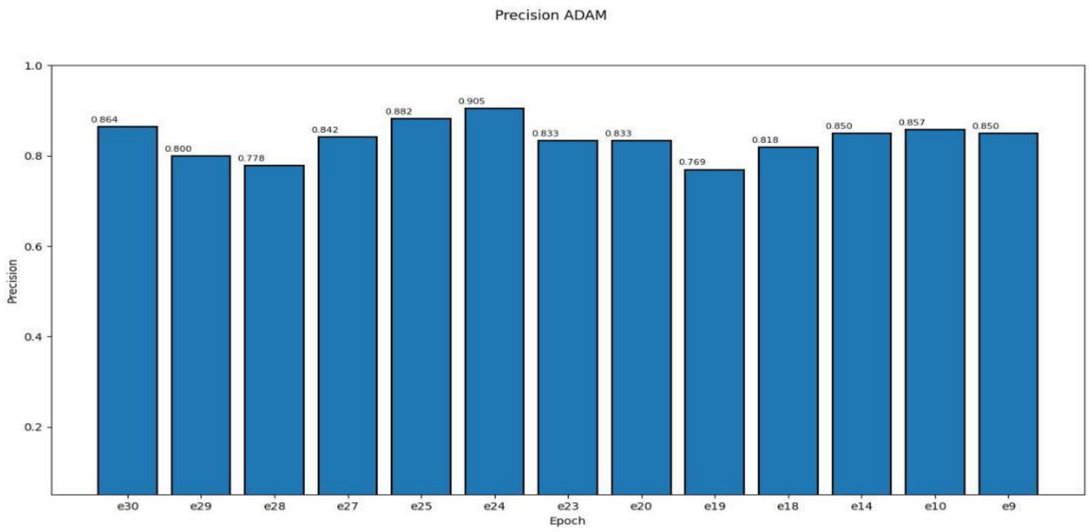
TP = *predict Abuse and actual result is Abuse*

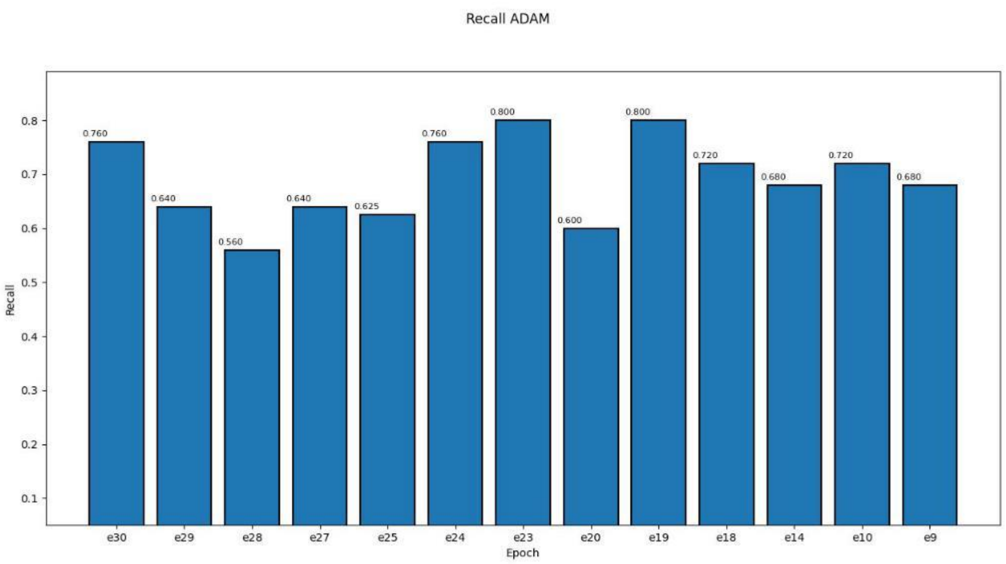
FN = *predict Not Abuse and actual result is Abuse*

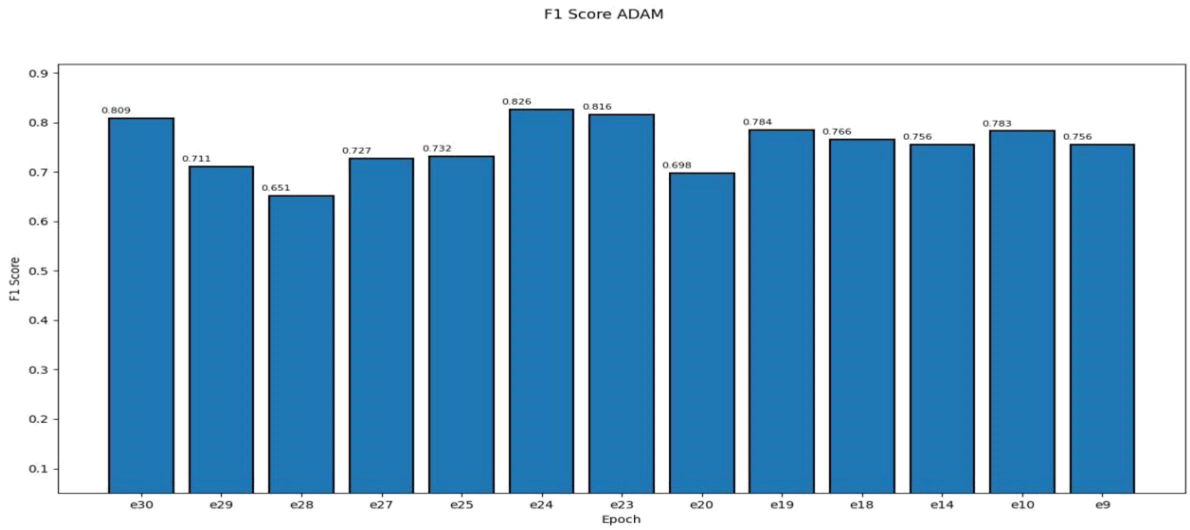
TN = *predict Not Abuse and actual result is Not Abuse*

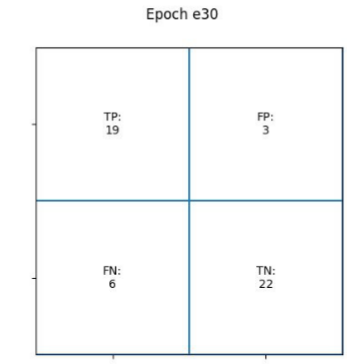
TP = *predict Abuse and actual result is Not Abuse*

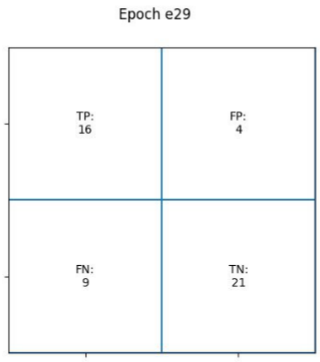
The following charts show the Precision, Recall, F1-score, confusion matrix result by the number of epoch and present all the evaluation metrics in one concluding table[Table -2]

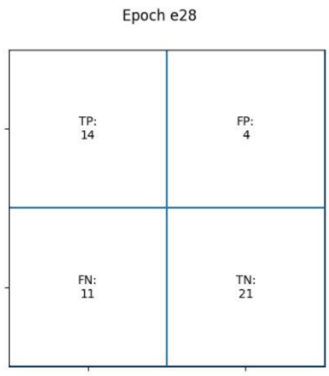


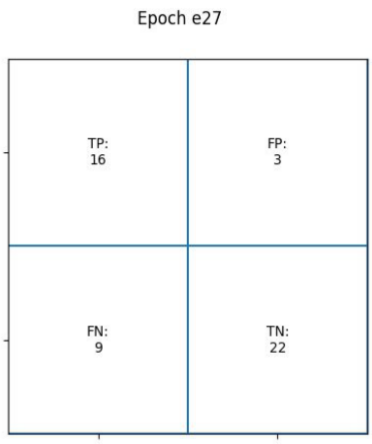


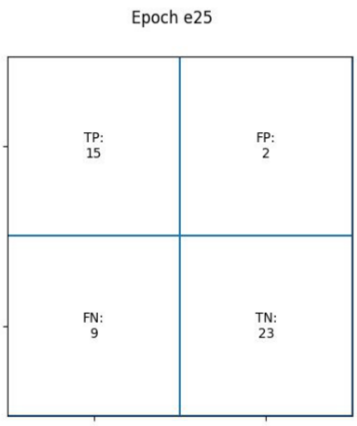

F1 Score ADAM
F1 Score
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.809
0.711
0.651
0.727
0.732
0.826
0.816
0.698
0.784
0.766
0.756
0.783
0.756
e30
e29
e28
e27
e25
e24
e23
e20
e19
e18
e14
e10
e9
Epoch

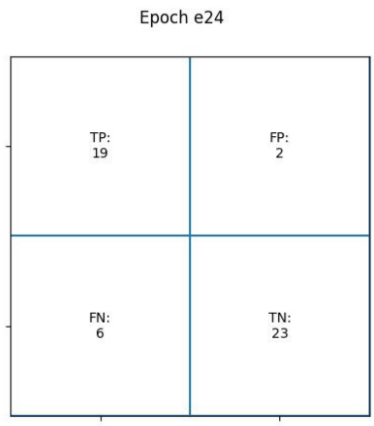

Epoch e30
TP:
19
FP:
3
FN:
6
TN:
22


Epoch e29
TP:
16
FP:
4
FN:
9
TN:
21

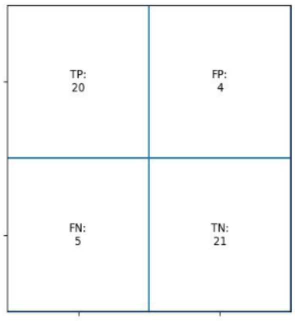

Epoch e28
TP:
14
FP:
4
FN:
11
TN:
21


Epoch e27
TP:
16
FP:
3
FN:
9
TN:
22

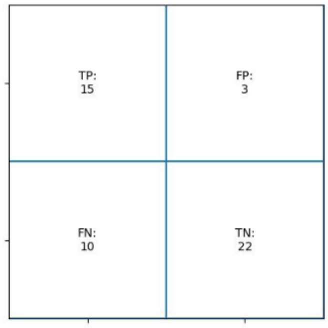

Epoch e25
TP:
15
FP:
2
FN:
9
TN:
23


Epoch e24
TP:
19
FP:
2
FN:
6
TN:
23

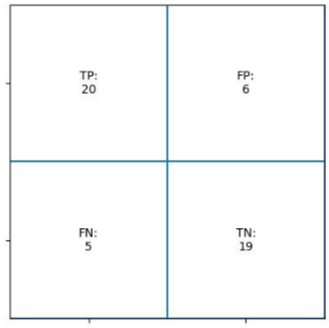
Epoch e23



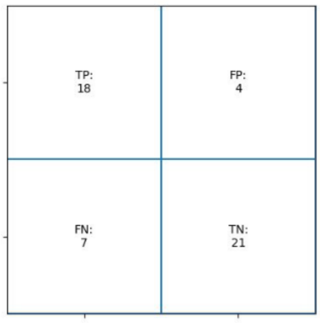
Epoch e20



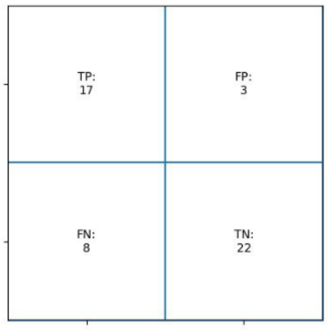
Epoch e19



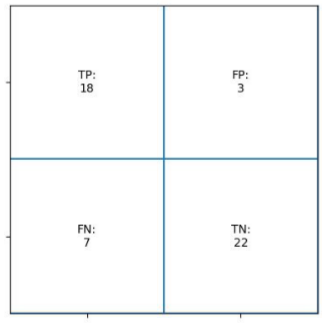
Epoch e18



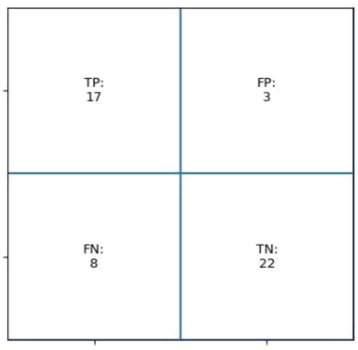
Epoch e14



Epoch e10



Epoch e9



we summarize the result as shown in Table 2

Table 2:result representation

mode at epoch	Train Accuracy	Val Accuracy	Test Accuracy	Recall	Precision	F1-Score
9	0.936	0.900	0.78	0.68	0.850	0.756
10	0.938	0.936	0.80	0.72	0.857	0.783
14	0.980	0.929	0.78	0.68	0.850	0.756
18	0.986	0.900	0.78	0.72	0.818	0.766
19	0.983	0.850	0.78	0.8	0.769	0.784
20	0.972	0.921	0.74	0.6	0.833	0.698
23	0.992	0.914	0.82	0.8	0.833	0.816
24	0.986	0.921	0.84	0.76	0.905	0.826
25	0.996	0.921	0.76	0.62	0.882	0.732
27	0.990	0.907	0.76	0.64	0.842	0.727
28	0.993	0.900	0.70	0.56	0.778	0.651
29	0.989	0.886	0.74	0.64	0.800	0.711
30	0.995	0.892	0.82	0.76	0.864	0.809

conclusions

As we can see in Table 3, the best performance was at epoch 24, and the model at epoch 24 got an**accuracy of 84%.**
The confusion matrix at epoch 24:

Table-3

TP 19 76%	FP 2 8%
FN 6 24%	TN 23 92%

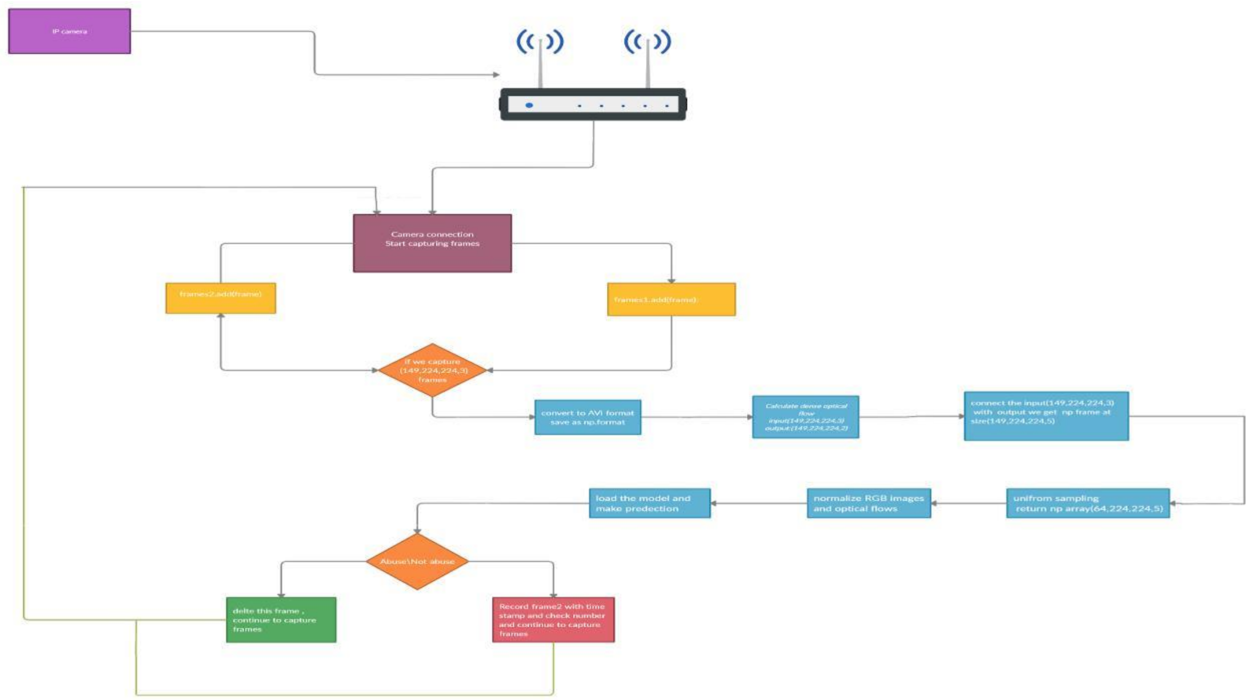
The Adam optimization algorithm has shown better performance on our test set than the previous SGD optimization algorithm.
We can see an improvement of 5% in F1-Score and an 8% improvement in recall.

mode at epoch	Train Accuracy	Val Accuracy	Test Accuracy	Recall	Precision	F1-Score
SGD epoch 19	0.89	0.829	0.8	0.68	0.89	0.77
ADAM epoch 24	0.986	0.921	0.84	0.76	0.905	0.826

Now, after we found the model with the best result, In order to gain deeper insights on the model performance in a real-time environment, the next step is to evaluate the model on our ADS prototype.

ADS evaluation

- **ADS structure** :
  frame1 = the frame that we will use to make a prediction - Abuse\Not abuse
  frame2 = the original frame format captured by the IP camera at the same time.



We built a data pipeline [Blue color] that executes the following every time when capturing 149 frames during broadcasting.

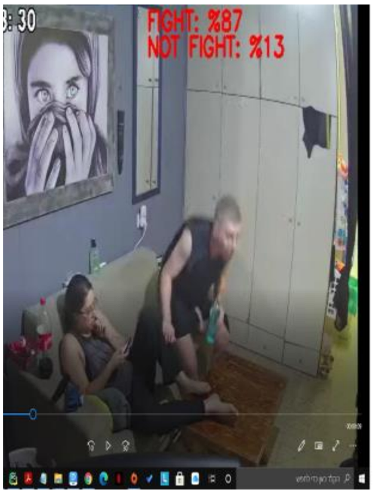
- convert all 149 frames to AVI format
- reshape frame size to (224,224)
- *Calculate dense optical flow input(149,224,224,3) output:(149,224,224,2)*
- connect the input(149,224,224,3) with output to get np frame of size(149,224,224,5)
- *uniform sampling return np array(64,224,224,5)*
- *normalize RGB images and optical flows*
- *load the model and make a prediction of Abuse /Not abuse on the np array(64,224,224,5)*

If the model prediction is Abuse then we will save frame2 as an Abuse event,
else we delete frame1 and frame2 and go back to capturing 149 frames.

In order to stimulate an adult abuse environment, we tested the following scenarios

- fast violence moment in ranging 5 seconds each.
- direct violence moment containing punching and kicking
- hidden violence moment when only the back of the attacker is seen
- moments where 2 people are far apart, but 1 person performs a sudden action [such as putting a cup of hot coffee on the table] that will cause a big optical change.

The following pictures show the video that the model classified as Abuse



ADS conclusions

The model will classify correctly violent events when accruing, and it only takes 4 seconds to make a prediction on np frame of size (149,224,224,5)

- The model classifies**video clips as violent**even when one person makes sudden action andthere is no proximity between the two people in the video, which actually makes a lot of sense, since the model is looking for changes in the optical spacing between 2 neighboring frames.
  Therefore any fast action, regardless of the distance between two people, will cause a significant change in the optical current.
  This is a very big problem for our system, because most of the nurses/caregivers in a nursing home perform additional activities on a daily basis, for example folding laundry, changing bedding, etc.

- The model has difficulty analyzing frames that contain a TV/computer screen.
  The sharp movement of the frames on the TV causes a high result in the optical current thus misleading the model and causing it to classify the event as abuse.
  Moreover, because in most nursing homes / private homes the camera is usually aimed at the living room and/or bedrooms that almost always have a TV, it can be misleading for the model.

Possible solution to discuss with academic mentor

In a broad look at the sub-tasks of computer vision, our project is a sub-task of action recognition. We will therefore look at other sub-tasks in computer vision to find solutions such as**Object Detection, Face Recognition, Visual Relationship Detection ect.**.

We thought about using the YOLO algorithm to detect objects. We can identify the position of the TV and by using coordinates we can blacken the area of the TV and thus solve the problem of TV/computer screens. In addition by using the YOLO algorithm, we can identify people in frames, and by using their position coordinates we can calculate the distance between 2 people so that we will run our model only when the distance between 2 people reaches a certain threshold.