Experiment report -1

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

Task definition

Our goals are to recognize abuse events in real-time through an IP camera,
using deep neural network architecture.
Today most computer vision recognition tasks use obvious choice Neural Network architecture such as 3DCNN, CNN, ConvLSTM, SVM.
Our demand for real-time detection forces us to consider the size of models. We want the smallest architecture that allows us to make predictions on the real-time system.
In most articles, we saw a combination of neural networks and handcrafted feature extraction like HOG, Trof, OpticalFlow, etc..
and on average the number of parameters is 24M-94M. That amount of parameters affects the prediction running time.
We found this article as most promising for our purpose

Ming Cheng, Kunjing Cai, and Ming Li. "RWF-2000: An Open Large Scale Video Database for Violence Detection." arXiv preprint arXiv:1911.05913 (2019).

The architecture proposed by the researchers was the smallest we could find that allowed us to make predictions in real-time.
This architecture has only 270,000 parameters and it has the highest accuracy compared to other models we saw.
In order to get an initial sense of the model behavior, and to see if it fits the task,
we load the pre-train model proposed by the researcher, **and got 68% accuracy** on our test abuse set.

Workspace

we use Vsat ai and rent a virtual machine
step 1: create virtual machines in https://vast.ai
step 2: install - docker image and Miniconda2-4.6.14-Linux-x86_64
step 3: install the following package
python version 3.6 , opencv=3.4.2 , keras=2.3.1, ipykernel ,
tensorflow-gpu=1.14.0 , cuda toolkit ,cudnn
step 4: download gsutil
connect to the bucket in the google cloud platform [GCP] and download the np file we filter.[Data Processing]

Data Processing

video Processing pipeline

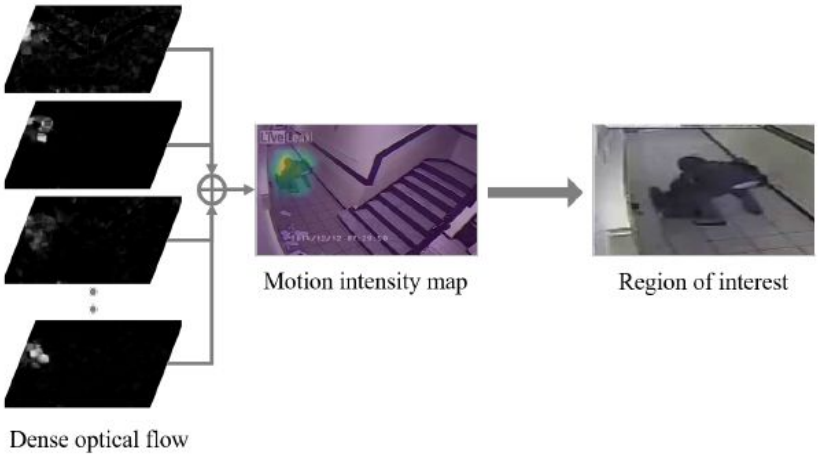
1. search abuse video links online
2. download the links
3. convert the video to AVI format with FBS=30sec
4. cut the video into 5-sec clips
5. manual extracting from each video 5sec clips [3,4 clips for each video]
6. create more videos by using 5 data argumentation techniques
7. split the data to Train, Val, Test as shown in table2

Data NP Generator pipeline

input: link to file containing 5sec video clips of size(149,224,224,5)
output: calculate the optical flow for each input and create np frame of size (149,224,224,5)

1. take 5 sec video clips frames of size(149,224,224,5)

2. calculate the **dense optical flow** for each frame[i],frame[i+1]
   and return flow frames of size (149,224,224,2)



3. combine the original frames with the flow to one variable result:
   result = flow[,,,,0:2] + orginal_frames[,,,:3]

4. save the result as .npy format

5. save all .npy to zip format

6. upload .npy videos to the google cloud platform

Data Set Description:

Abuse of the elderly:
We downloaded more than 200 videos through the pipeline[video_preproccisng.py] and after manually sorting them we extracted 160 video clips of elder abuse.

Child abuse
We downloaded more than 150 videos through the pipeline[video_preproccisng.py] and after manually sorting them we extracted 70 video clips of child abuse, most of the video clips are from CCTV

Street Fight
This data set is from **NTU CCTV-Fights Dataset** and contains 1000 videos.
We extracted 250 fight clips that were taken using a mobile phone camera.

Adult care
We downloaded more than 200 videos through the pipeline[video_preproccisng.py] and after manually sorting them we extracted 175 video clips that contain actions such as changing a diaper, dressing, feeding, nursing, lifting, etc..

Normal behavior_1
We downloaded more than 150 videos through the pipeline[video_preproccisng.py] and after manually sorting them we extracted 100 video clips of normal behavior such as lifting a sofa, moving furniture, walking at home, etc..
We search for routine activities that take place at home regularly.

Normal behavior_2
This data set is from the **Real Life Violence Dataset.** We extracted 150 video clips of normal behavior such as people eating, talking, lifting things and human contact.

Table 1: data set partition

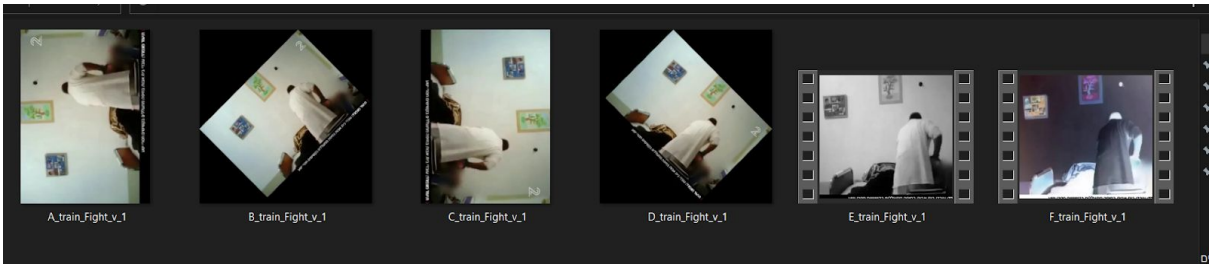
source	Train	Val	Test
Abuse of the elderly	120	20	25
child abuse	50	10	0
Street Fight	156	40	0
Adult care	127	20	25
Normal behavior_1	100	0	0
Normal behavior_2	100	50	0

We split the data as shown in **table 2,** we collected 842 video clips after the filtering process. It is difficult to gather abuse videos due to censorship and inappropriate content on the Internet.
In addition, the majority of abuse videos online are mostly edits and are full of advertisements for the news channels.

Table 2: Abuse / Not Abuse distribution

	Abuse	Not Abuse	Total
Train	326	326	652
Val	70	70	140
Test	25	25	50

Our solution to the lack of quality data was to use 6 data augmentation techniques which change video clips to--> [90,-90,45,-45 degree , black_white color , invert color]
as shown in the picture below.



We implement these techniques only on the Training set and the clips of elder abuse.
The final data set is shown in Table3

Table 3: Abuse / Not Abuse final distribution

	Abuse	Not Abuse	Total
Train	2282	2282	4564
Val	70	70	140
Test	25	25	50

Experiment hyperparameter

The research used **Stochastic gradient descent (SGD) with Momentum.**
We used the same learning rate, learning decay and momentum.

Table 5:experiment hyperparameter

learning rate α	**0.01**
learning decay	**1e-6**
momentum β	**0.9**
batch size	**6**
Number of workers	**6**
Number of epoch	**30**
GPU	1x Tesla V100

Model architecture

We used the same architecture that the researchers suggested: Conv3D split into two networks, one for RGB frame and one for Optical flows, as shown in the figure below

Table II
PARAMETERS OF THE MODEL ARCHITECTURE (THE *t* REPRESENTS THE NUMBER OF REPEATS)

Block Name	Type	Filter Shape	t
RGB/Flow Channels	Conv3d	1×3×3@16	
	Conv3d	3×1×1@16	2
	MaxPool3d	1×2×2	
	Conv3d	1×3×3@32	
	Conv3d	3×1×1@32	2
	MaxPool3d	1×2×2	
Fusion and Pooling	Multiply	None	1
	MaxPool3d	8×1×1	1
Merging Block	Conv3d	1×3×3@64	
	Conv3d	3×1×1@64	2
	MaxPool3d	2×2×2	
	Conv3d	1×3×3@128	
	Conv3d	3×1×1@128	1
	MaxPool3d	2×2×2	
Fully-connected Layers	FC layer	128	2
	Softmax	2	1

Result

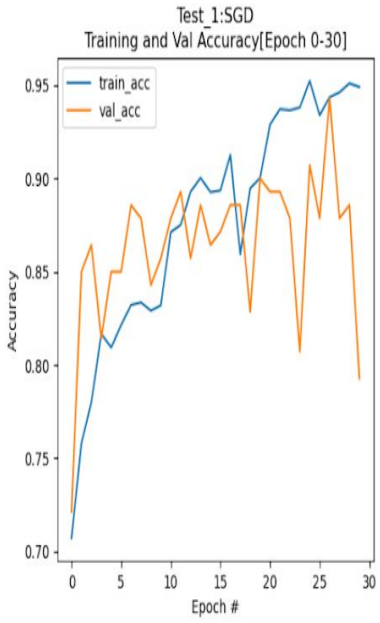
Training result

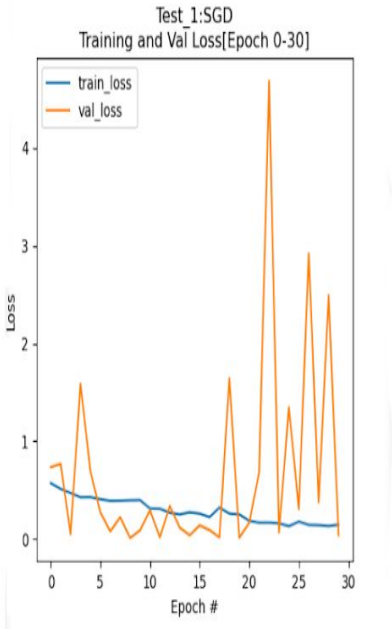
We search for the minimum gap between Test error and val error

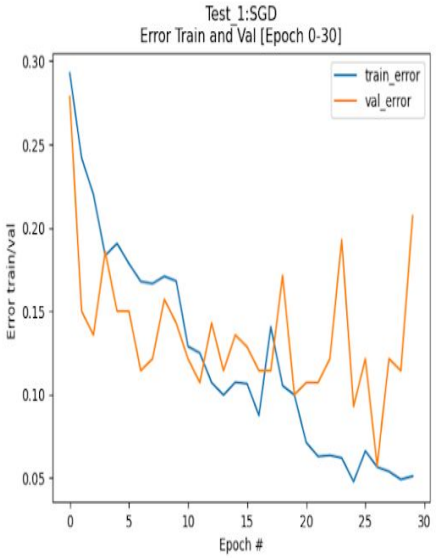
$$Test\ Error\ = 1 - \ Test\ Accuracy$$

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

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



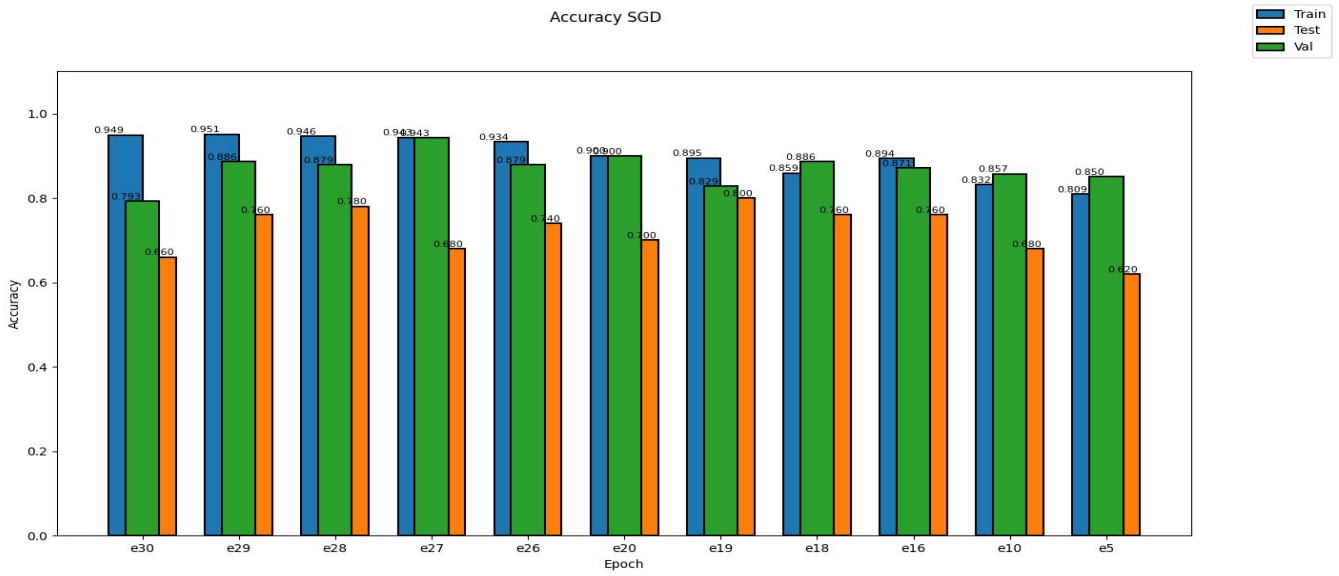


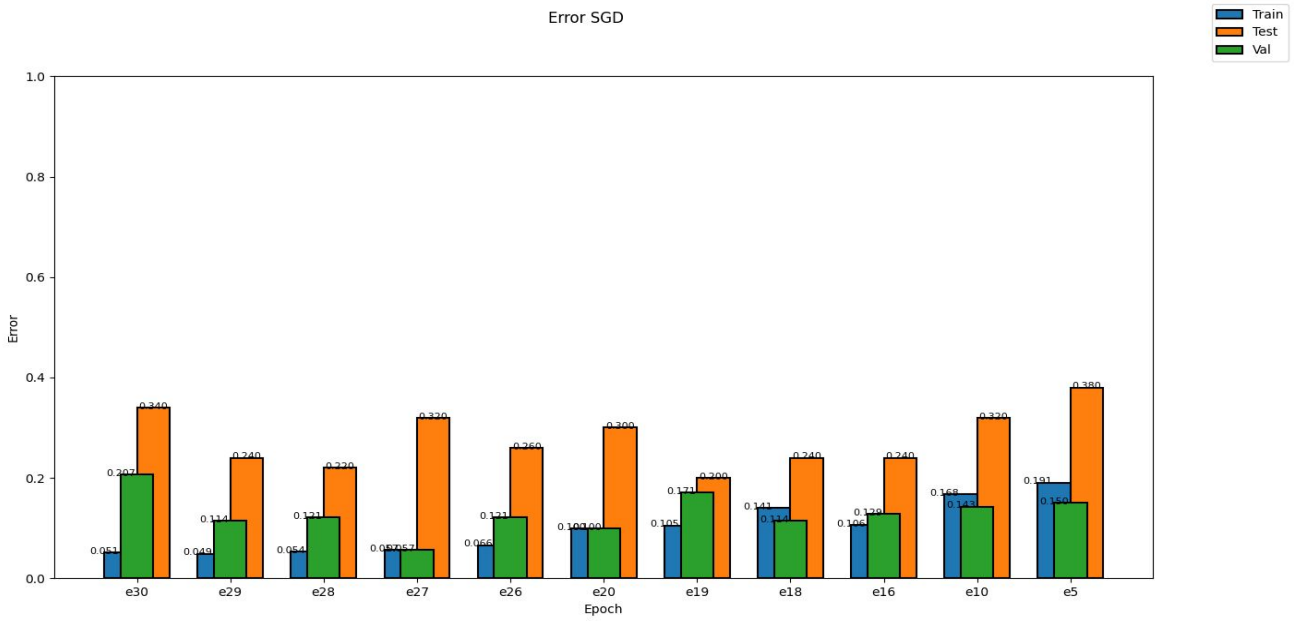


We selected the models with the lowest error rate and highest accuracy on the dev set.
We found the following models to be the most promising, for testing on our test set:
models at epoch number [30,29,28,27,26,20,19,18,16,10].

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

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





Evaluation metrics

We used the following metrics Accuracy, Precision, recall, F1-score

$$Recall = \frac{TP}{TP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

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

$$TP = predict\ Abuse\ and\ actual\ result\ is\ Abuse$$

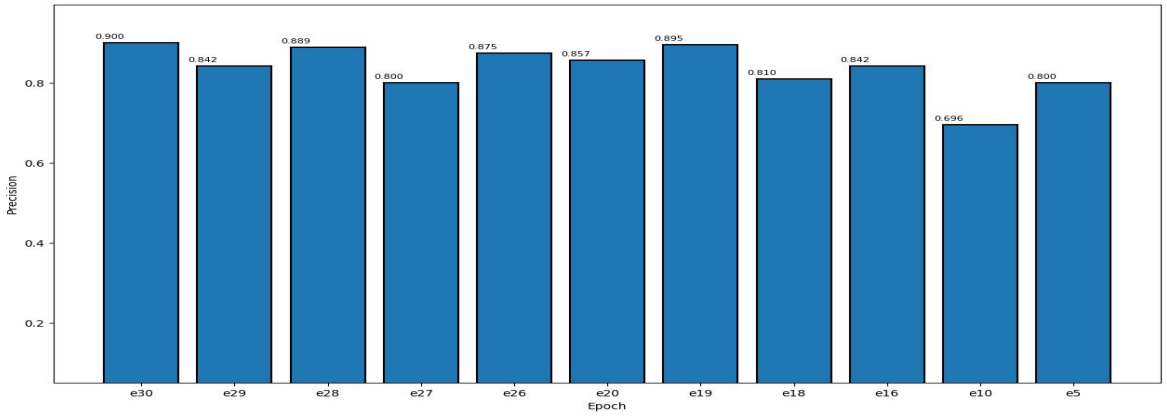
$$TN = predict\ Not\ Abuse\ and\ actual\ result\ is\ Not\ Abuse$$

$$FN = predict\ Not\ Abuse\ and\ actual\ result\ is\ Abuse$$

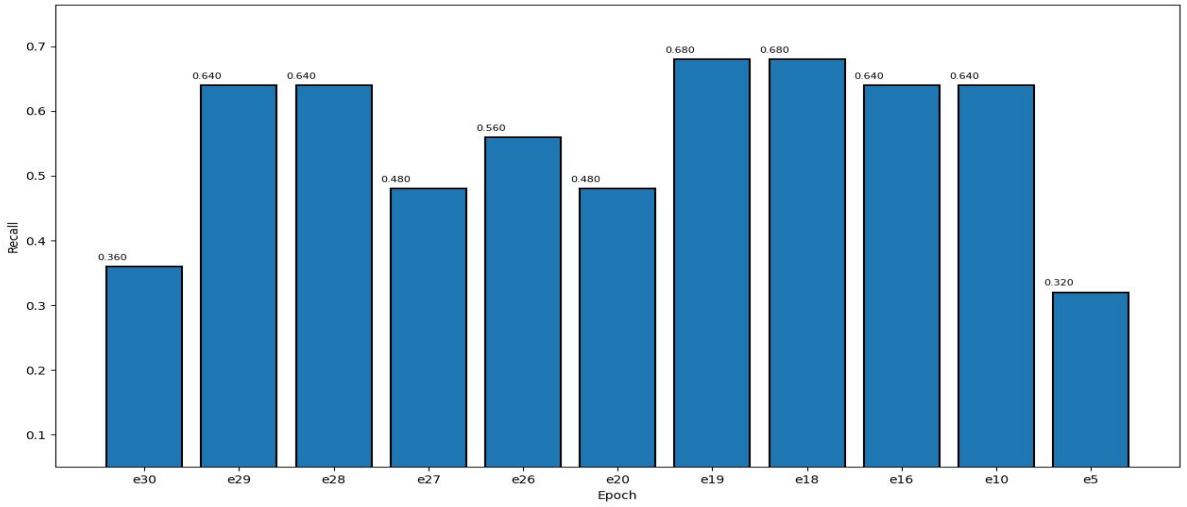
$$TP = predict\ Abuse\ and\ actual\ result\ is\ Not\ Abuse$$

The following charts show the Precision, Recall, F1-score, confusion matrix results by the number of epoch and present all the evaluation metrics in one concluding table[Table -6]

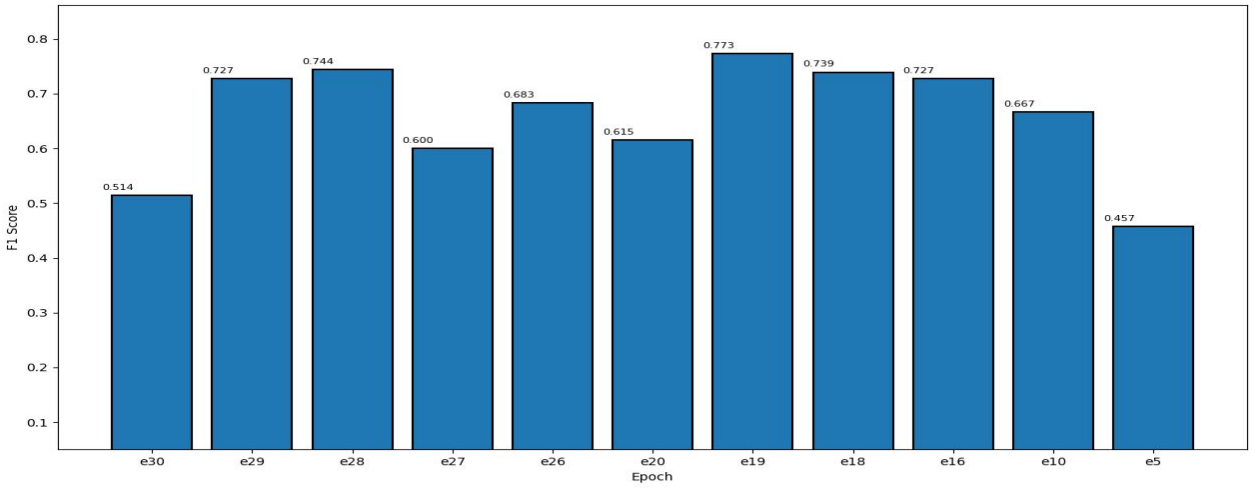
Precision SGD



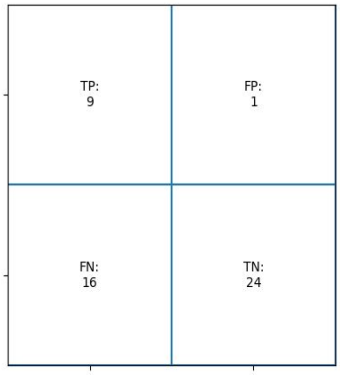
Recall SGD



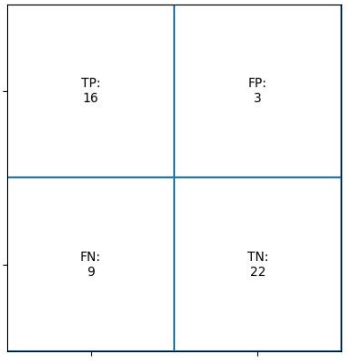
F1 Score SGD



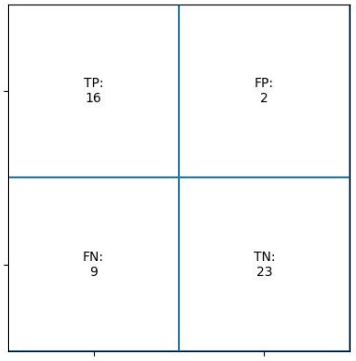
Epoch e30



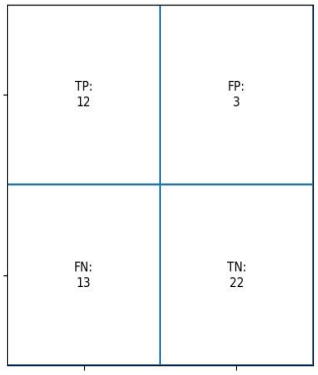
Epoch e29



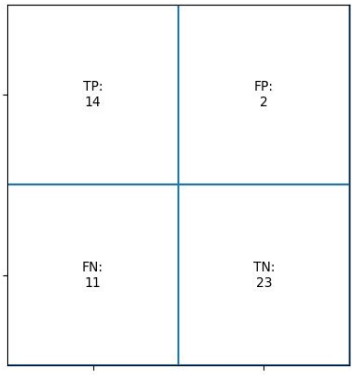
Epoch e28



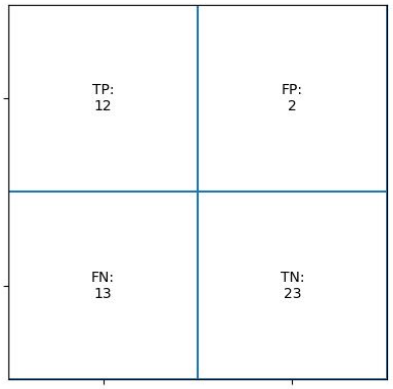
Epoch e27



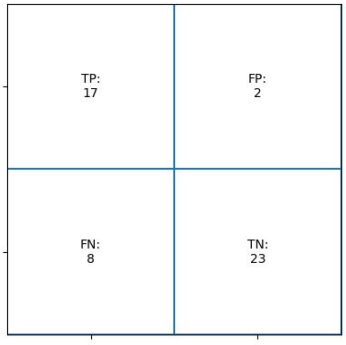
Epoch e26



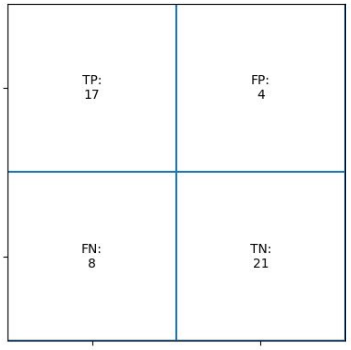
Epoch e20



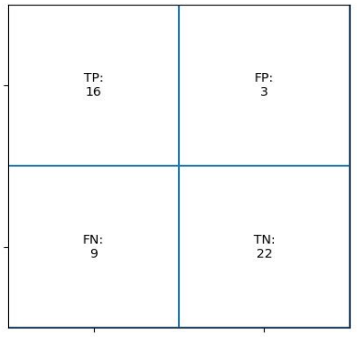
Epoch e19



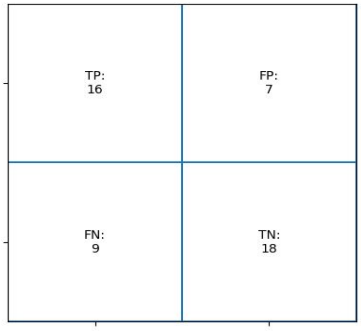
Epoch e18



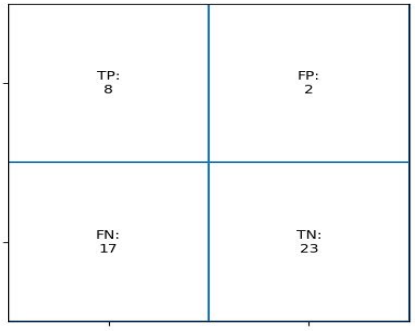
Epoch e16



Epoch e10



Epoch e5



We summarized the results as shown in Table 6

Table 6:result representation

mode at epoch	Train Accuracy	Val Accuracy	Test Accuracy	Recall	Precision	F1-Score
5	0.81	0.850	0.62	0.32	0.46	0.46
10	0.83	0.857	0.68	0.64	0.70	0.67
16	0.89	0.871	0.76	0.64	0.84	0.73
18	0.86	0.886	0.76	0.68	0.81	0.74
19	0.89	0.829	0.8	0.68	0.89	0.77
20	0.90	0.900	0.7	0.48	0.86	0.62
26	0.93	0.879	0.74	0.56	0.88	0.68
27	0.94	0.943	0.68	0.48	0.80	0.60
28	0.95	0.879	0.78	0.64	0.89	0.74
29	0.95	0.886	0.76	0.64	0.84	0.73
30	0.95	0.793	0.66	0.36	0.90	0.51

Conclusions

As we can see in Table 6 the best performance was at epoch 19,
and as mentioned [Taks definition] we got on the pre-train model **accuracy of 68%** on our abuse test set and the model at epoch 19 got an **accuracy of 80%**
the confusion matrix at epoch 19:

TP 17 68%	FP 2 8%
FN 8 32%	TN 23 92%

From 25 video clips in the test set that we manually classified as abuse of the elderly video clips, the model predictions were right in **68%.**
The model predictions for 25 video clips that we manually classified
as not abuse videos **[see Data Set Description Adult care data set]** was right in **92%.**
These results are encouraging because our test set is very hard in terms of video clips chosen as not abuse. We want the model to distinguish between adult care cases and abuse cases.

For example, we want the model to distinguish between proper hoisting and support for the adults by nurses, doctors, and therapists who usually come in close contact with the patient **[Most often they are the ones who abuse adults],**
therefore the model would have difficulty classifying them as an Abuse\NotAbuse event.

In the next experiment, we mean to:

- Search for other optimization methods that will improve performance on our test set.
- build a prototype that will be tested on the best performing model in a real-time environment for better evaluations.