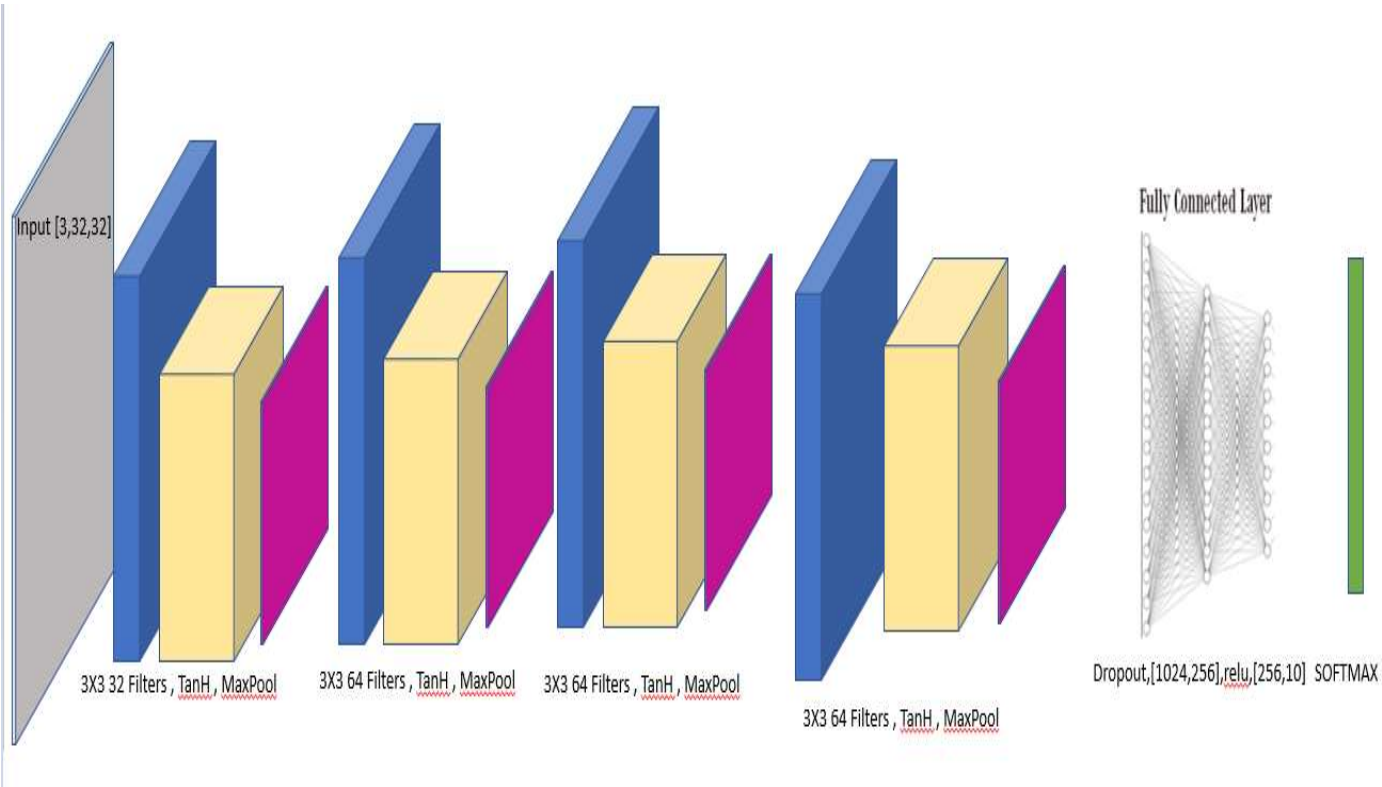
We've Implemented a CNN based on the following visualization:




```
conv = Conv(in_channels=3, out_channels=32, kernel_size=3, stride=1, padding=1)
conv2 = Conv(in_channels=32, out_channels=64, kernel_size=3, stride=1, padding=1)
conv3 = Conv(in_channels=64, out_channels=64, kernel_size=3, stride=1, padding=1)
fc1 = Linear(in_size=1024, out_size=256, uniform_range=0.1)
fc2 = Linear(in_size=256, out_size=10, uniform_range=0.1)
model = CNN_Model(
    layers=[conv, TanH(), MaxPool(filter_size=2), conv2, TanH(), MaxPool(filter_size=2), conv3, TanH(),
            MaxPool(filter_size=2), Flatten(), Dropout(p=0.5), fc1, LinearRelu(), fc2, Softmax()],
```


Yuval Cohen 205883499, Oriel Asraf 313453540

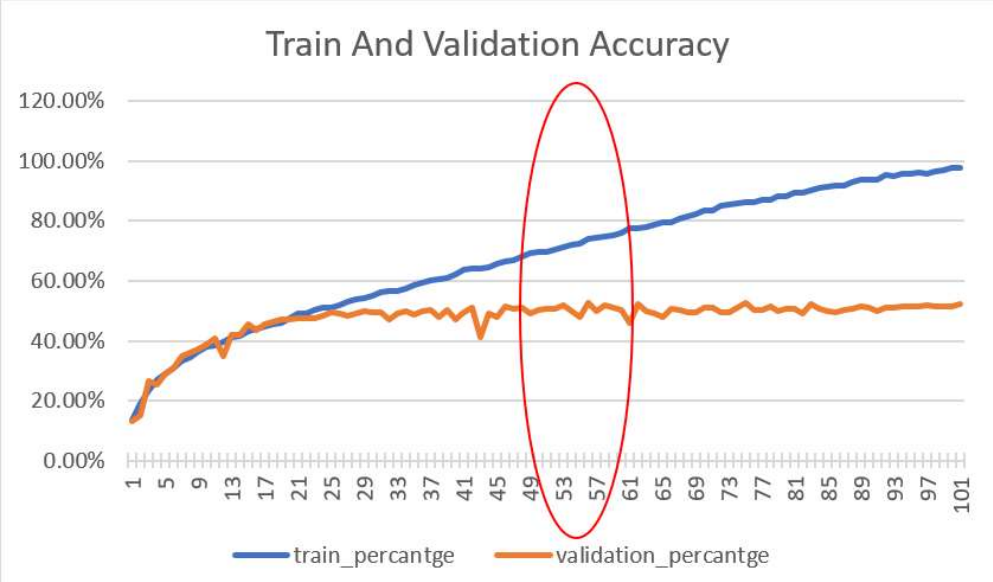
Optimization – SGD with batch size of 100

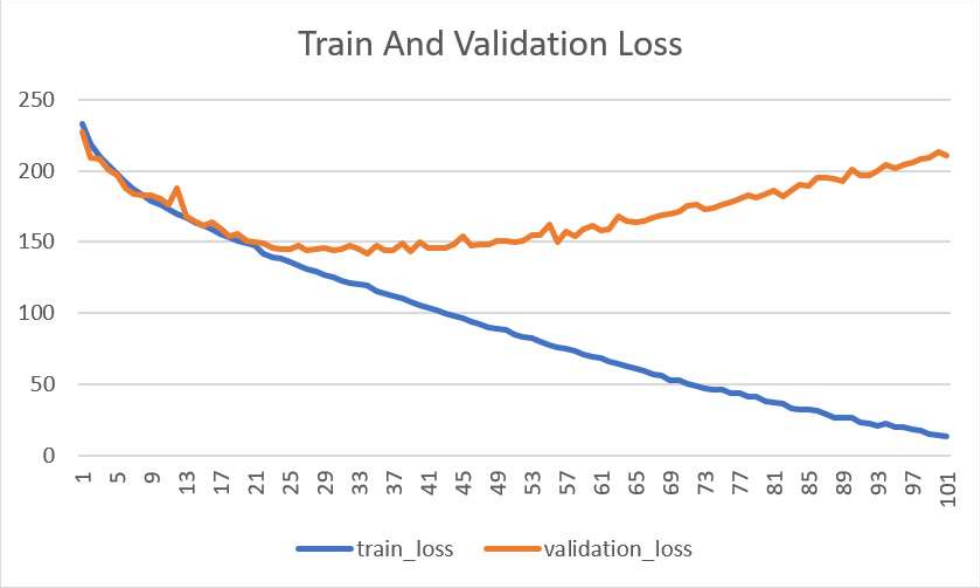
Learning rate - 0.035

Overall, in our best case our model got **52.8%** accuracy on the validation data set:

epochs = 56

(circled in red – where we concluded was the best part to stop training) *





*We've worked really hard on this project and preformed multiple experiments, we've also implemented GEM(col2im and im2col) for fast Convolution, Unfortunately we had a few setbacks(in real life) that made us compromise on 52-53 % (and also not show these experiments)

Instructions:

In order to run the pretrained model , you should run the following command:

"python trained.py [name of pretrained model] [name of test data]"

For example:

"**python trained.py trained_model.sav test.csv**" *

Test.csv should be in the same directory as main.py

*Notice that this specific args are the default for this program so it would work with

"**python trained.py**" as well.

If you want to retrain the model, you should run

"python main.py" *

*Notice that 'train.csv and validate.csv' must be in the same directory (while training it output percentage and loss every epoch.)

After the program completes , a new file is generated named "trained_model_new.sav"

And of course you can use it with trained.py like in this shown example:

"python trained.py trained_model_new.sav [name of test data.csv]"