VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Computer Science and Engineering

J Component report

Programme : B.Tech(CSE- AIR)

Course Title : Fundamentals of Artificial Intelligence

Course Code : CSE2039

Slot : F1

Title: Melanoma Detection

Team Members: Kavya Paliwal | 20BRS1111

Aneesh Aparajit | 20BRS1054

Hritik Goel | 20BRS1035

Aniket Kumar Paul | 20BRS1116

Faculty: Dr. R. Jothi **Sign:** R Jothi

Date: 29/4/22

MELANOMA DETECTION

Kavya Paliwal (20BRS1111), Aneesh Aparajit G (20BRS1054), Aniket Kumar Paul (20BRS1116), Hritik Goel (20BRS1035)

VELLORE INSTITUE OF TECHNOLOGY CHENNAI
VANDALUR – KELAMBAKKAM ROAD
CHENNAI - 600127

Abstract*—Melanoma, most threatening type of skin cancer, is on the rise. In this paper an implementation of a deep-learning system on a computer server, equipped with graphic processing unit (GPU), is proposed for detection of melanoma lesions. Clinical (non-dermoscopic) images are used in the proposed system, which could assist a dermatologist in early diagnosis of this type of skin cancer. In the proposed system, input clinical images, which could contain illumination and noise effects, are preprocessed in order to reduce such artifacts. Afterward, the enhanced images are fed to a pre-trained convolutional neural network (CNN) which is a member of deep learning models. The CNN classifier, which is trained by large number of training samples, distinguishes between melanoma and benign cases. Experimental results show that the proposed method is superior in terms of diagnostic accuracy in comparison with the state-of-the-art methods.

I. INTRODUCTION

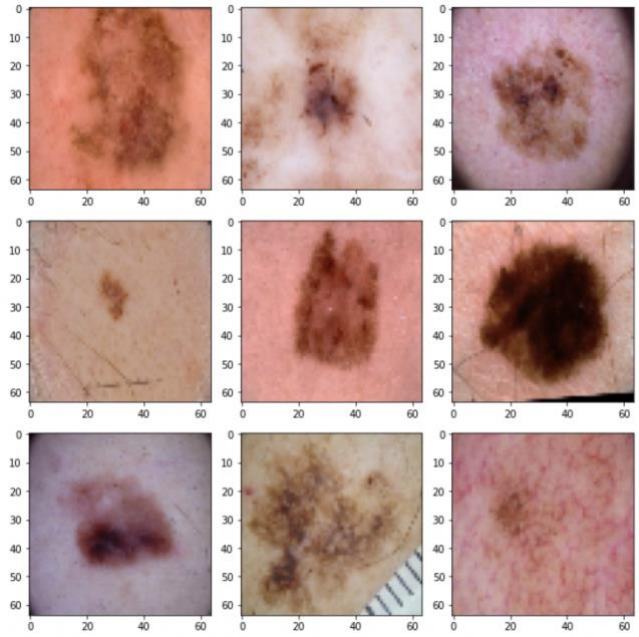
Melanoma, also referred to as malignant melanoma, is a type of skin cancer caused by abnormal multiplication of pigment producing cells that give color to the skin. Despite significant death rate, early stage detected melanoma is curable in most cases. Meanwhile differentiation between melanoma and other benign moles in their initial growth phases is a challenging task even for experienced dermatologists. Computerized algorithms are being developed for this purpose. Some low complexity methods are designed, which are intended for running on tablets and smartphones, and can help non-specialists. But professional decision making, in this regard, requires sophisticated algorithms and equipment. There are various methods in dermatology such as ABCD (asymmetry, border irregularity, color patterns, and diameter) rule and the seven-point checklist that guide physicians in this task. Figure 1 shows the general structure of our proposed, highly complex, deep learning method.

Automated analysis of pigmented lesions is a growing research topic that aims to develop tools for computer aided diagnosis of skin cancer [6]. Reviews of researches done in this field are given in [6] and [7]. Recently there has been an emerging trend for automatic diagnosis of melanoma using conventional digital cameras [8-14]. This can be applicable in web based and mobile application as a telemedicine tool and also as a supporting system that assists physicians. Computerized diagnosis is a rising need due to increasing rate of incident, subjectivity of procedure and time and cost expenses [11]. A decision support system was proposed by Alcon et al. that takes advantage of image processing techniques for evaluation of skin lesions [8]. It also uses some background information of patient before the diagnosis is made. Cavalcanti et al. proposed an automatic classification system that consists of preprocessing, segmentation and feature extraction steps. In order to reduce miss rate of melanoma cases (false negative cases), a two stage classifier has been proposed that rechecks the lesions labeled as benign [10]. Deep learning approaches have shown promising results in some applications such as natural language processing, speech recognition and recently in computer vision areas such as object tracking, object detection, and image classification. Deep learning mechanisms can learn set of high level features from low level ones and gain high accuracy for classification applications without the need for extracting handcrafted features. Especially there has been a trend for taking advantage of superior power of deep learning methods in medical imaging tasks. Convolutional neural network (CNN) is a type of deep learning method where trainable filters and pooling operations are applied on the raw input images, extracting set of complex high level features automatically. In this paper we aim to take advantage of deep learning methods to form an automatic diagnosis system for melanoma detection. For this purpose the input digital images, usually subjected to noise and illumination effects, are preprocessed. This preprocessing effectively helps CNN extract discriminative features from the images. Afterward, the images are fed to a CNN architecture to classify the input as melanoma or benign. In order to deal with the limitations of the training dataset and relatively low number of images, some approaches such as rotation, cropping, and resizing of dataset images are considered that increase the number of samples. The experimental results show that the proposed system can outperform comparable state-of-the-art methods. The rest of this paper is organized as follow. In Section II, the proposed method is explained in details. Experimental results of quantitative evaluation of our method are presented in Section III. Section IV concludes the paper.

II. DATASET

Datasets used in this research were from Kaggle. SIIM-ISIC Melanoma Classification

Link: https://www.kaggle.com/c/siim-isic-melanoma-classification
Skin Cancer: Malignant vs. Benign
Link: https://www.kaggle.com/fanconic/skin-cancer-malignant-vs-benign
Skin Cancer MNIST - HAM10000
Link: https://www.kaggle.com/kmader/skin-cancer-mnist-ham10000



III. MODEL ARCHITECTURE

Since, this is a medicine related dataset, we must not focus on the accuracy, but on reducing the number of False Negatives. That is, say a person has skin cancer, but we predict that he/she doesn't have cancer. This is dangerous because he/she may lose his life. But, if we predict that the person has skin cancer even though he/she doesn't although it's a mis-classification, it can be prefered more because, he/she will get it tested and a life may be saved.

So, the metric should be metrics like *False_Negative*, *False_Positive*, *Precision* and *Recall* rather than *Accuracy* in this case.

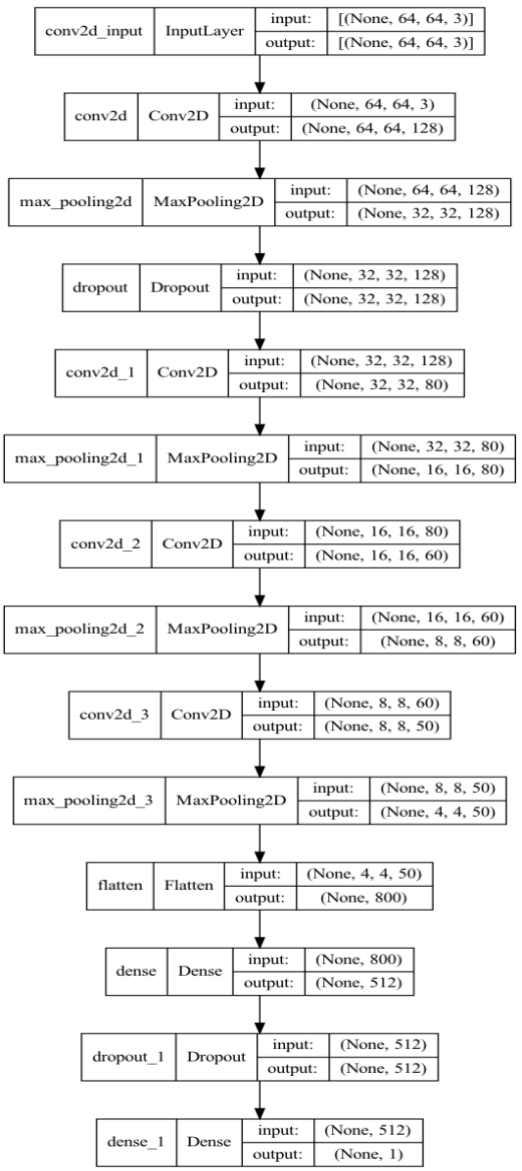


Fig 1: Model

A. 32x32 CNN with no Data Augmentation

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 32, 32, 3)]	0
cnn (CNN)	(None, 32, 32, 64)	1088
max_pooling2d (MaxPooling2D)	(None, 16, 16, 64)	0
dropout (Dropout)	(None, 16, 16, 64)	0
cnn_1 (CNN)	(None, 16, 16, 40)	23240
max_pooling2d_1(MaxPooling2D)	(None, 8, 8, 40)	0
cnn_2 (CNN)	(None, 8, 8, 30)	10950
max_pooling2d_2(MaxPooling2D)	(None, 4, 4, 30)	0

cnn_3 (CNN)	(None, 4, 4, 25)	6875
max_pooling2d_3 (MaxPooling2D)	(None, 2, 2, 25)	0
flatten (Flatten)	(None, 100)	0
dense (Dense)	(None, 512)	51712
dropout_1 (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 1)	513

Total params: 94,378
Trainable params: 94,060
Non-trainable params: 318

B. 64x64 CNN with no Augmentation

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 64, 64, 3)]	0
cnn (CNN)	(None, 64, 64, 128)	2176
max_pooling2d (MaxPooling2D)	(None, 32, 32, 128)	0
dropout (Dropout)	(None, 32, 32, 128)	0
cnn_1 (CNN)	(None, 32, 32, 80)	92560
max_pooling2d_1(MaxPooling2D)	(None, 16, 16, 80)	0
cnn_2 (CNN)	(None, 16, 16, 60)	43500
max_pooling2d_2(MaxPooling2D)	(None, 8, 8, 60)	0
cnn_3 (CNN)	(None, 8, 8, 50)	27250
max_pooling2d_3 (MaxPooling2D)	(None, 4, 4, 50)	0
flatten (Flatten)	(None, 800)	0
dense (Dense)	(None, 512)	410112
dropout_1 (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 1)	513

Total params: 576,111
Trainable params: 575,475
Non-trainable params: 636

C. 128x128 CNN with no Augmentation

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 32, 32, 3)]	0
cnn (CNN)	(None, 32, 32, 192)	2176
max_pooling2d (MaxPooling2D)	(None, 16, 16, 192)	0
dropout (Dropout)	(None, 16, 16, 192)	0
cnn_1 (CNN)	(None, 16, 16, 120)	92560
max_pooling2d_1(MaxPooling2D)	(None, 8, 8, 120)	0
cnn_2 (CNN)	(None, 8, 8, 90)	43500
max_pooling2d_2(MaxPooling2D)	(None, 4, 4, 90)	0
cnn_3 (CNN)	(None, 4, 4, 75)	27250
max_pooling2d_3 (MaxPooling2D)	(None, 2, 2, 75)	0
flatten (Flatten)	(None, 300)	0
dense (Dense)	(None, 512)	410112
dropout_1 (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 1)	513

Total params: 522,896
Trainable params: 521,942
Non-trainable params: 954

D. 32x32 with Augmented Data

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 32, 32, 3)]	0
cnn (CNN)	(None, 32, 32, 64)	1088
max_pooling2d (MaxPooling2D)	(None, 16, 16, 64)	0
dropout (Dropout)	(None, 16, 16, 64)	0
cnn_1 (CNN)	(None, 16, 16, 40)	23240
max_pooling2d_1(MaxPooling2D)	(None, 8, 8, 40)	0
cnn_2 (CNN)	(None, 8, 8, 30)	10950
max_pooling2d_2(MaxPooling2D)	(None, 4, 4, 30)	0
cnn_3 (CNN)	(None, 4, 4, 25)	6875
max_pooling2d_3 (MaxPooling2D)	(None, 2, 2, 25)	0
flatten (Flatten)	(None, 100)	0
dense (Dense)	(None, 512)	51712
dropout_1 (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 1)	513

Total params: 94,378
Trainable params: 94,060
Non-trainable params: 318

```
CNN is a utility class which has a
covolutional layer and a batch normalization
layer.
class CNN(layers.Layer):
  def __init__(self, filters, kernel_size, strides, padding):
    super(CNN, self).__init__()
    self.conv = layers.Conv2D(
      filters=filters,
      kernel_size=kernel_size,
      strides=strides,
      padding=padding,
      activation='relu'
    )
    self.bn = layers.BatchNormalization()

  def call(self, x, training=False):
    x = self.conv(x)
    x = self.bn(x)
    return tf.nn.relu(x)
```

E. 64x64 with Augmented Data

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 64, 64, 3)]	0
cnn (CNN)	(None, 64, 64, 128)	2176
max_pooling2d (MaxPooling2D)	(None, 32, 32, 128)	0
dropout (Dropout)	(None, 32, 32, 128)	0
cnn_1 (CNN)	(None, 32, 32, 80)	92560
max_pooling2d_1(MaxPooling2D)	(None, 16, 16, 80)	0
cnn_2 (CNN)	(None, 16, 16, 60)	43500
max_pooling2d_2(MaxPooling2D)	(None, 8, 8, 60)	0
cnn_3 (CNN)	(None, 8, 8, 50)	27250
max_pooling2d_3 (MaxPooling2D)	(None, 4, 4, 50)	0
flatten (Flatten)	(None, 800)	0
dense (Dense)	(None, 512)	410112
dropout_1 (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 1)	513

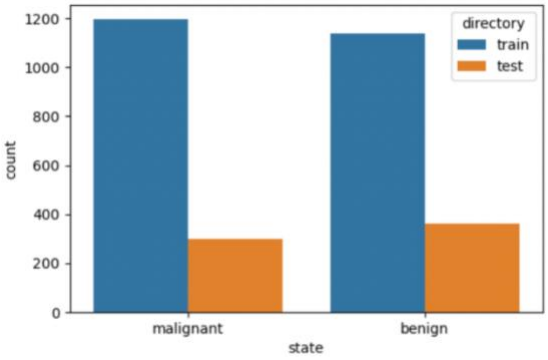
```
Total params: 576,111
Trainable params: 575,475
Non-trainable params: 636
```

IV. AUGMENTATION

- Before Augmentation

Dataset	No. of Images
Train	2374
Validation	263
Test	660

Class	No. of Images	Directory
Malignant	1197	Train
Benign	1140	Train
Malignant	300	Test
Benign	360	Test

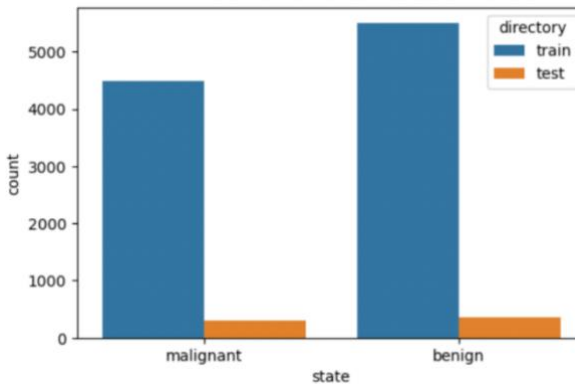


- After Augmentation

```
import Augmentor as ag
p = ag.Pipeline("../data/train/")
p.rotate90(probability=0.5)
p.zoom(probability=0.3, min_factor=0.8,
max_factor=1.5)
p.flip_left_right(probability=0.8)
p.flip_top_bottom(probability=0.3)
p.random_brightness(probability=0.3, min_factor=0.3,
max_factor=1.2)
p.sample(10000)
```

Dataset	No. of Images
Train	9001
Validation	999
Test	660

Class	No. of Images	Directory
Malignant	4497	Train
Benign	5503	Train
Malignant	300	Test
Benign	360	Test



V. TRANSFER LEARNING

Transfer Learning is the technique used to train a Convolutional Neural Network (CNN) to determine chance of a mole to be cancerous based on a dermatological photo. The way transfer learning works is that instead of initializing all of a neural network's weights to some random value and then learn image feature representation from scratch, you resuse the feature representation learned by a neural network on another dataset. In most cases what people use are benchmark DNN (Deep Neural Network) which have performed very well on the ImageNet dataset which contains a 1000 categories with a thousand image in each of those categories.

A. Transfer Learning on Augmented Data

Architecture

Layer (type)	Output Shape	Param #
input_2 (InputLayer)	[(None, 64, 64, 3)]	0
resnet101v2 (Functional)	(None, None, None, 2048)	42626656 0
global_average_pooling2d (GlobalAveragePooling2D)	(None, 2048)	0
dense (Dense)	(None, 2)	4098

```
Total params: 42,630,658
Trainable params: 42,532,994
Non-trainable params: 97,664
```

VI. FRAMEWORKS USED

- Tensorflow
- FastAPI
- Augmentor
- Numpy
- Pandas
- Matplotlib
- PIL
- Uvicorn

VII. RESULTS

A. 32x32 CNN with no Data Augmentation

Train Data

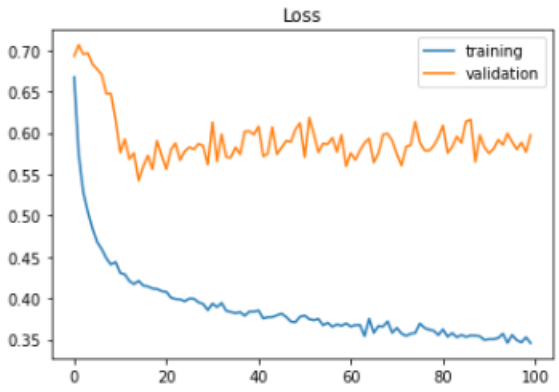
Metric	Score
loss	0.4670059084892273
auc	0.9204818606376648
false_negative	516.0
false_positives	65.0
precision	0.8963317275047302
recall	0.5213358402252197
accuracy	0.7552654147148132

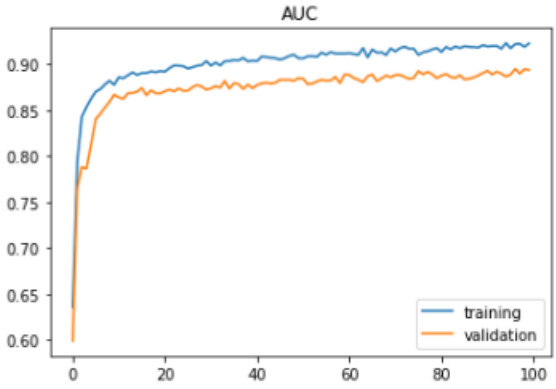
Validation Data

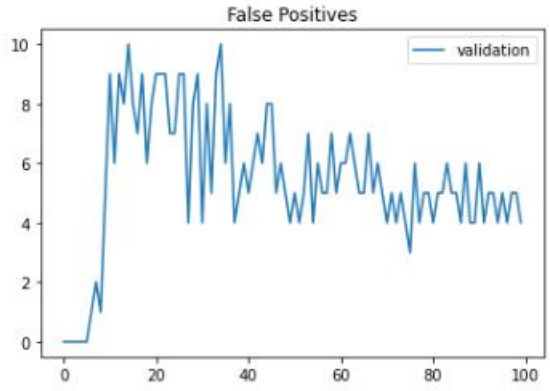
Metric	Score
loss	0.6094912886619568
auc	0.886875569820404
false_negative	79.0
false_positives	5.0
precision	0.8888888955116272
recall	0.3361344635486603
accuracy	0.680608332157135

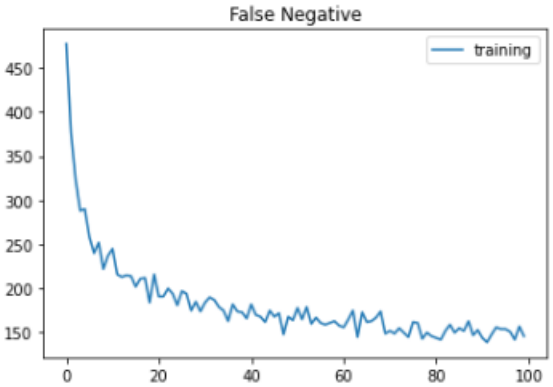
Test Data

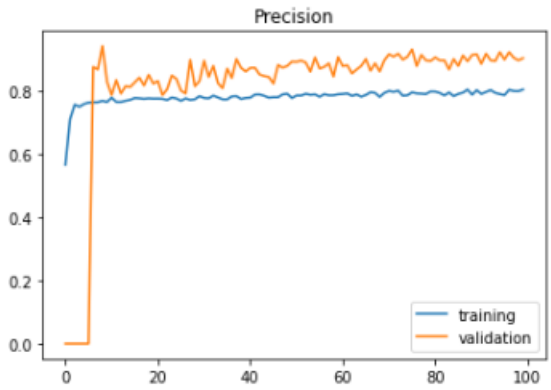
Metric	Score
loss	0.46654874086380005
auc	0.9121388792991638
false_negative	140.0
false_positives	20.0
precision	0.8888888955116272
recall	0.5333333611488342
accuracy	0.7575758099555969

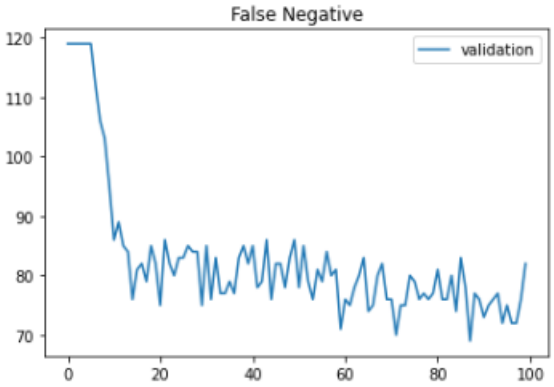


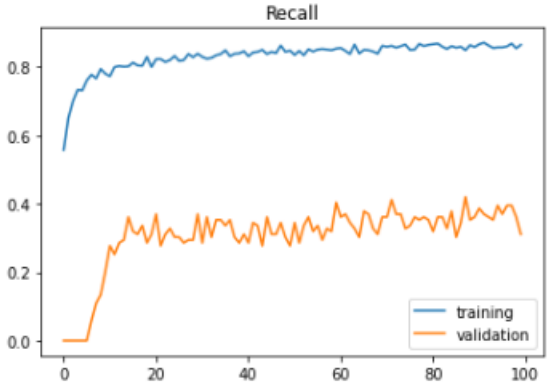


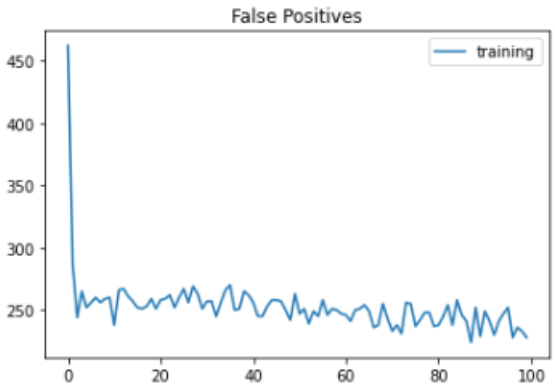


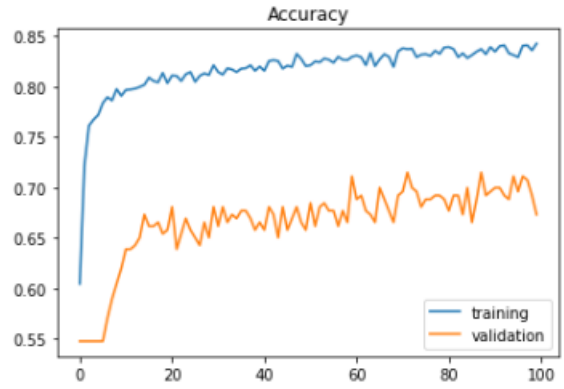












B. 64x64 CNN with no Augmentation

Train Data

Metric	Score
loss	0.7700790166854858
auc	0.9124453663825989

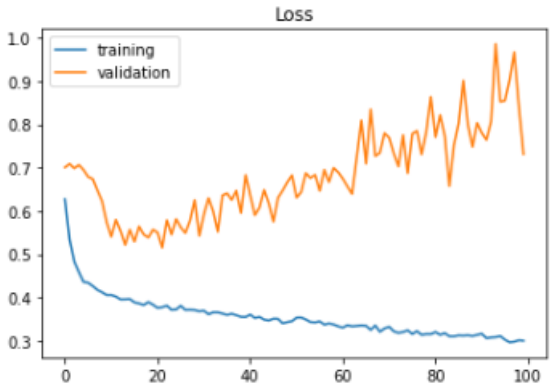
false_negative	923.0
false_positives	4.0
precision	0.9748427867889404
recall	0.1437847912311554
accuracy	0.6095198392868042

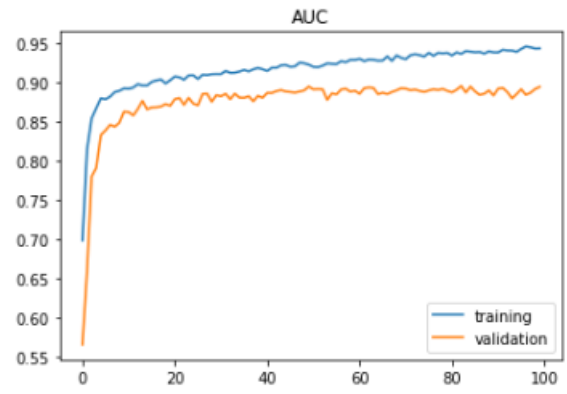
Validation Data

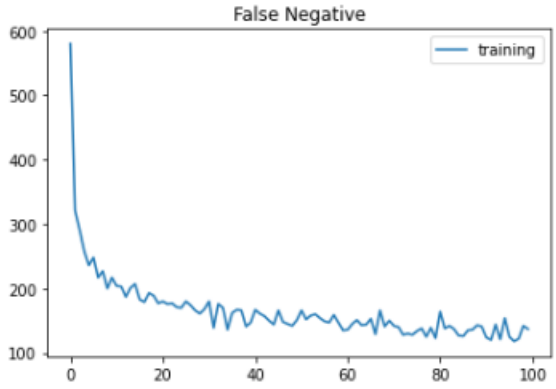
Metric	Score
loss	0.937388002872467
auc	0.8764297962188721
false_negative	107.0
false_positives	0.0
precision	1.0
recall	0.10084034502506256
accuracy	0.5931558609008789

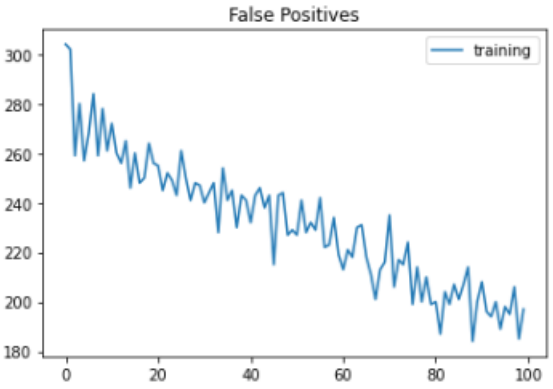
Test Data

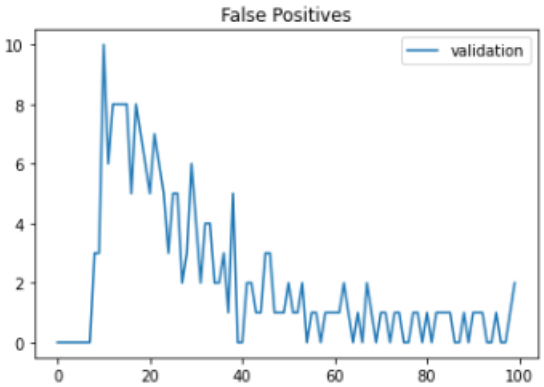
Metric	Score
loss	0.783999502658844
auc	0.8971203565597534
false_negative	267.0
false_positives	1.0
precision	0.9705882668495178
recall	0.10999999940395355
accuracy	0.5939394235610962

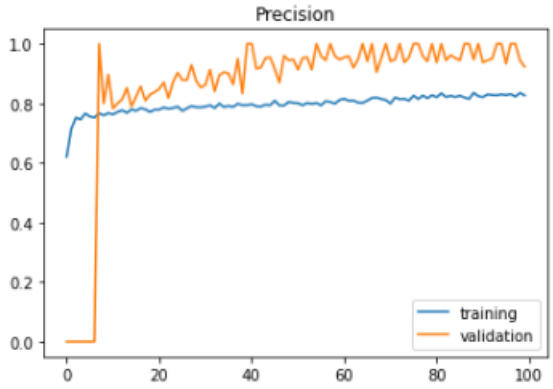


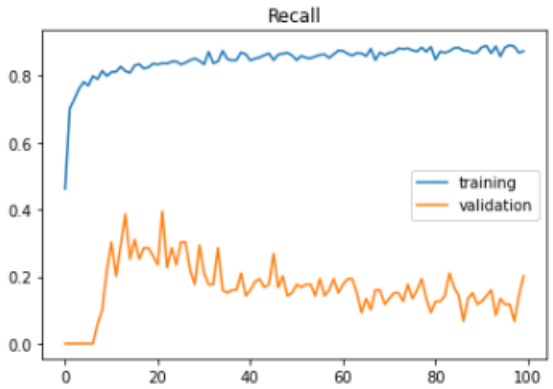


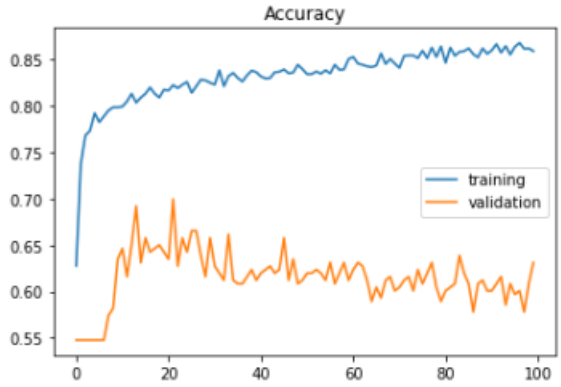












C. 128x128 CNN with no Augmentation

Train Data

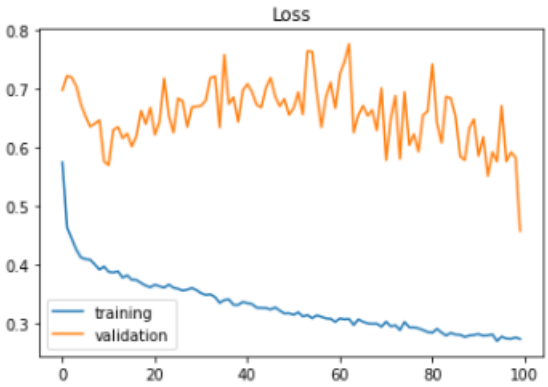
Metric	Score
loss	0.5062731504440308
auc	0.9309614300727844
false_negative	602.0
false_positives	27.0
precision	0.9463221430778503
recall	0.44155845046043396
accuracy	0.73504638671875

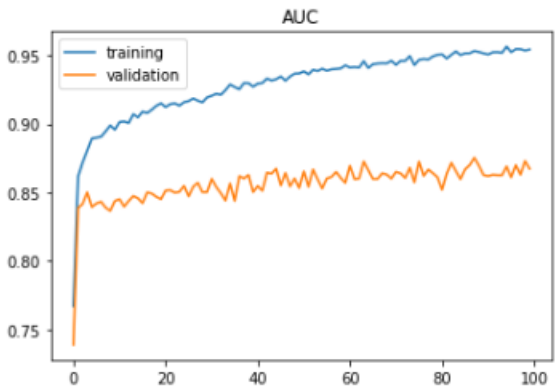
Validation Data

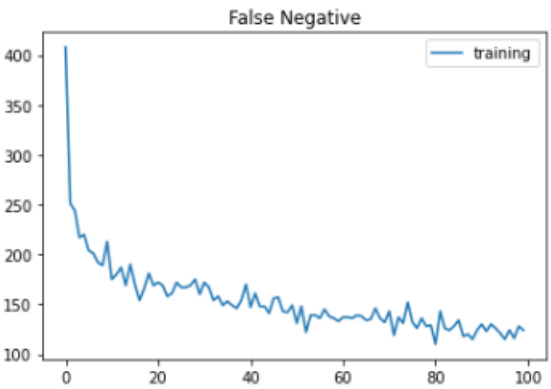
Metric	Score
loss	0.6643155217170715
auc	0.8909897804260254
false_negative	86.0
false_positives	3.0
precision	0.9166666865348816
recall	0.27731093764305115
accuracy	0.661596953868866

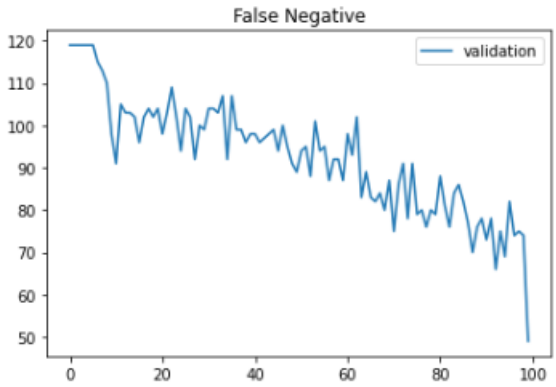
Test Data

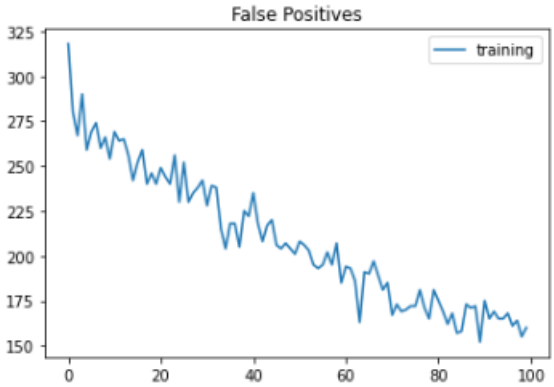
Metric	Score
loss	0.5401652455329895
auc	0.9101296067237854
false_negative	169.0
false_positives	10.0
precision	0.9290780425071716
recall	0.43666666746139526
accuracy	0.728787899017334

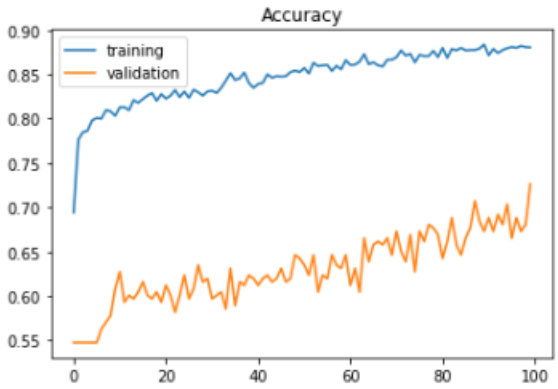


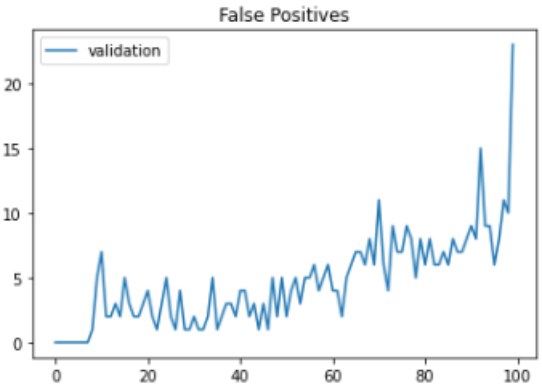


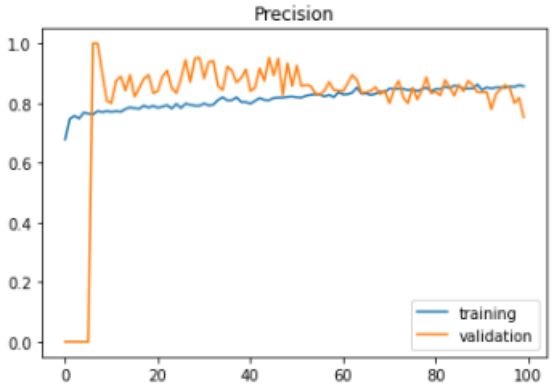












D. 32x32 with Augmented Data

Train Data

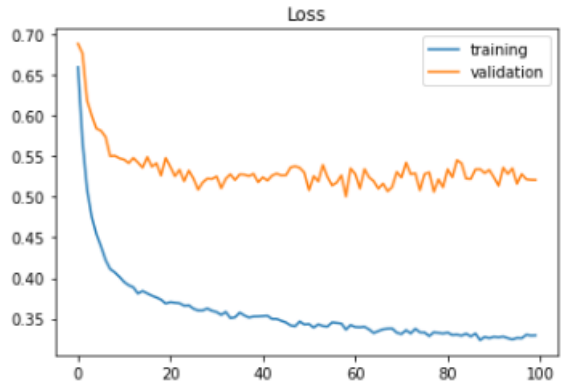
Metric	Score
loss	0.37595465779304504
auc	0.935274600982666
false_negative	1547.0
false_positives	242.0
precision	0.9117754697799683
recall	0.6178359985351562
accuracy	0.80124431848526

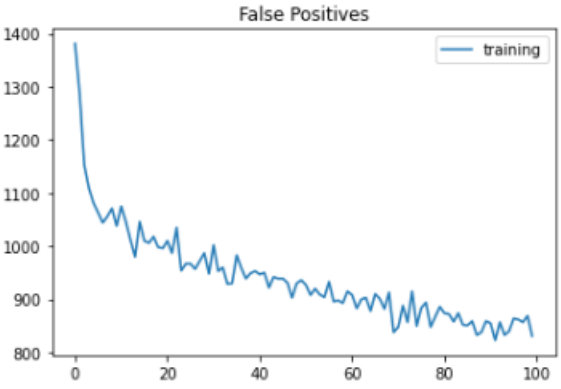
Validation Data

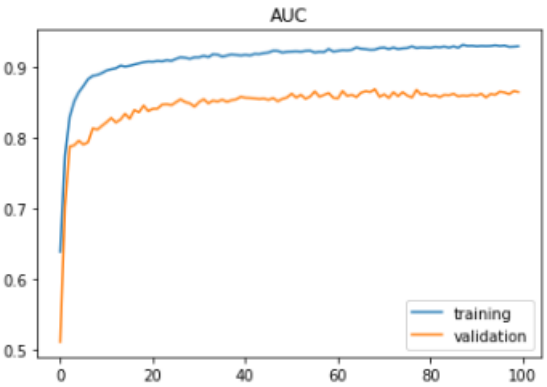
Metric	Score
loss	0.5271267294883728
auc	0.8663373589515686
false_negative	257.0
false_positives	35.0
precision	0.8458150625228882
recall	0.4276169538497925
accuracy	0.7077077031135559

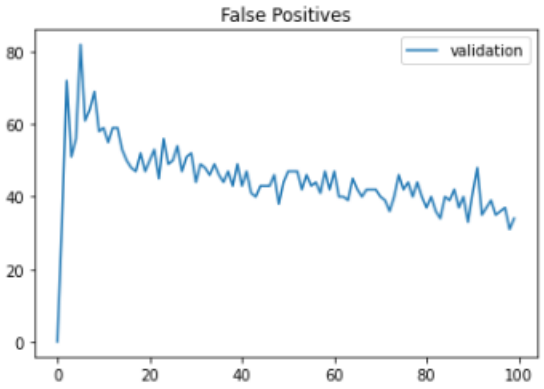
Test Data

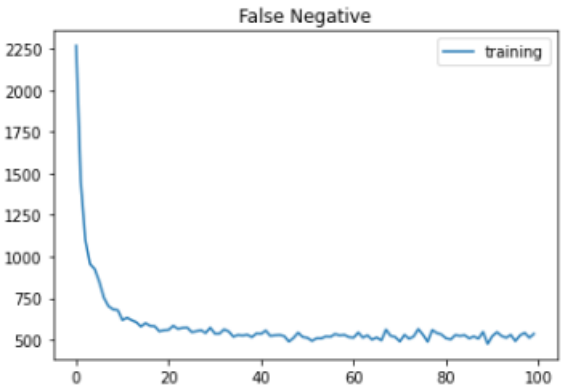
Metric	Score
loss	0.4034781754016876
auc	0.9088425636291504
false_negative	99.0
false_positives	32.0
precision	0.8626609444618225
recall	0.6700000166893005
accuracy	0.8015151619911194

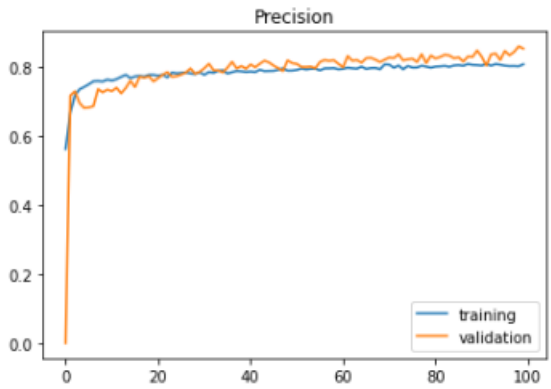

Loss
training
validation

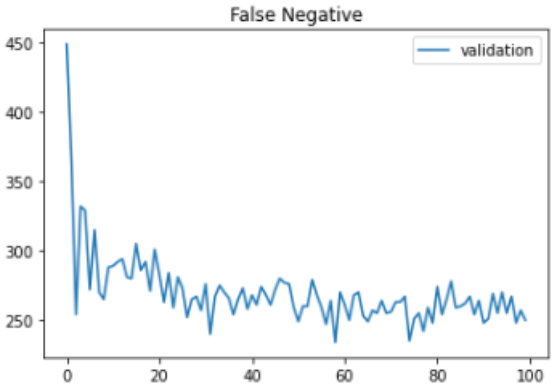

False Positives
training

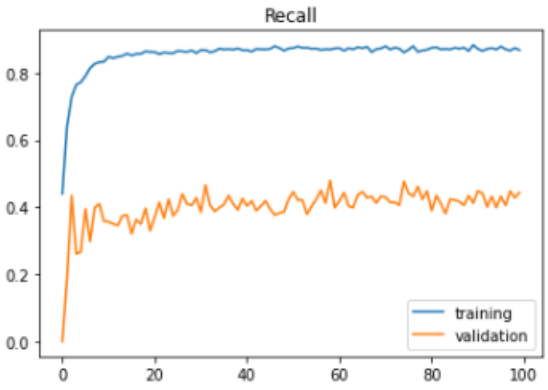

AUC
training
validation

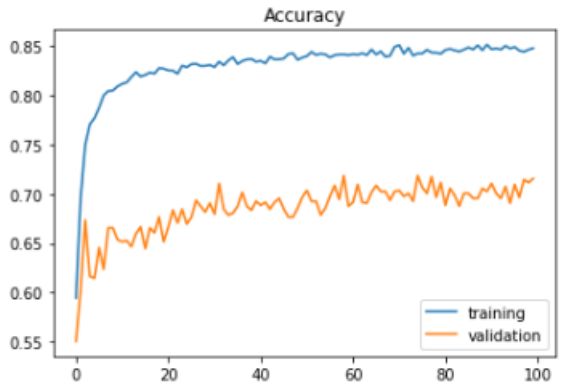

False Positives
validation

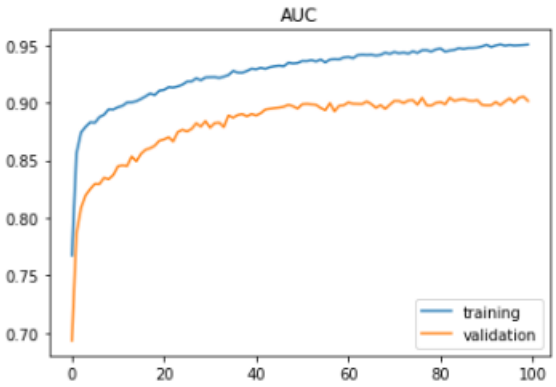

False Negative
training


Precision
training
validation


False Negative
validation


Recall
training
validation

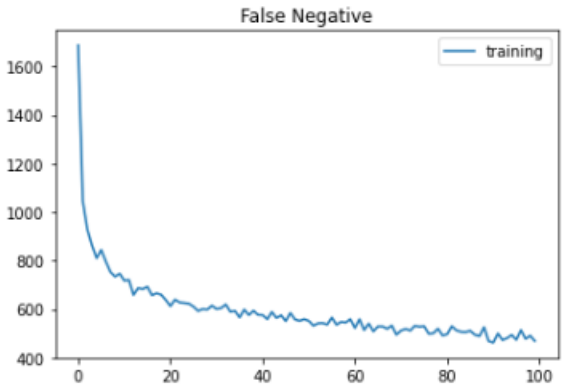




E. 64x64 with Augmented Data

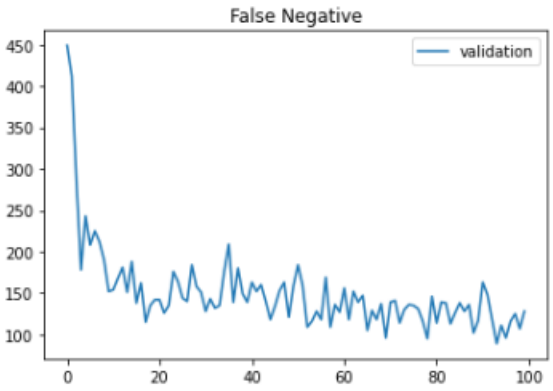
Train Data

Metric	Score
loss	0.3086279034614563
auc	0.9400833249092102
false_negative	698.0
false_positives	588.0
precision	0.8506856560707092
recall	0.8275691866874695
accuracy	0.8571269512176514



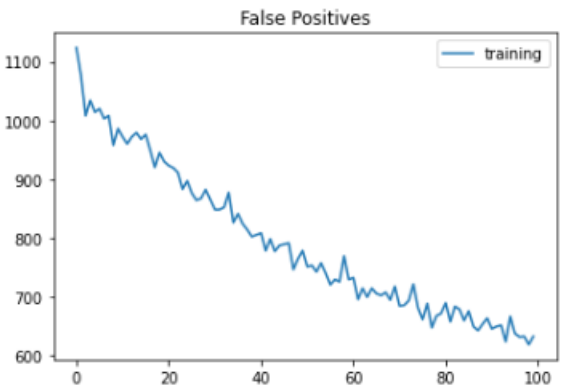
Validation Data

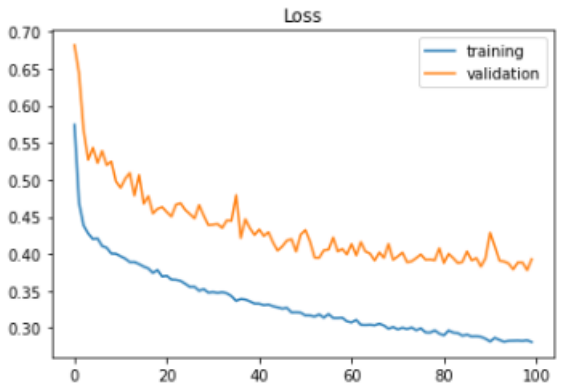
Metric	Score
loss	0.39539089798927307
auc	0.8995059728622437
false_negative	131.0
false_positives	80.0
precision	0.7989949584007263
recall	0.7082405686378479
accuracy	0.7887887954711914

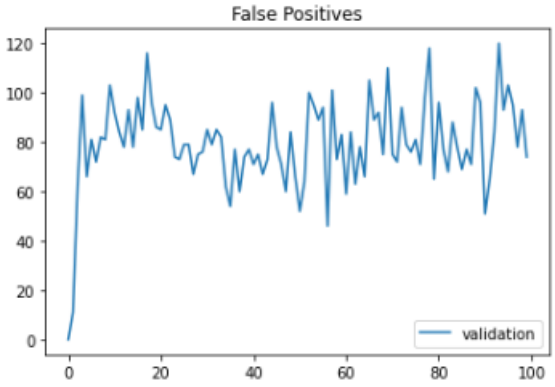


Test Data

Metric	Score
loss	0.3531557321548462
auc	0.9238147735595703
false_negative	29.0
false_positives	72.0
precision	0.7900875210762024
recall	0.9033333659172058
accuracy	0.846969723701477



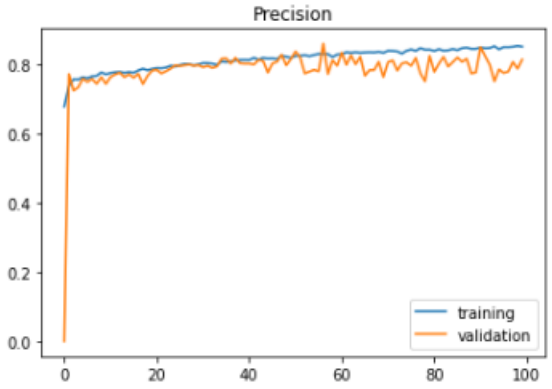




loss	0.006930578034371138
auc	0.9999805688858032
false_negative	4.0
false_positives	4.0
precision	0.9995555877685547
recall	0.9995555877685547
accuracy	0.9995555877685547

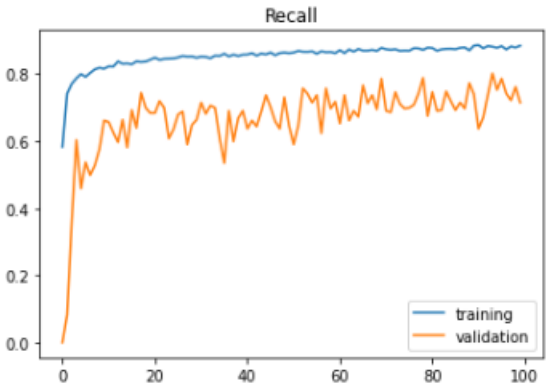
Validation Data

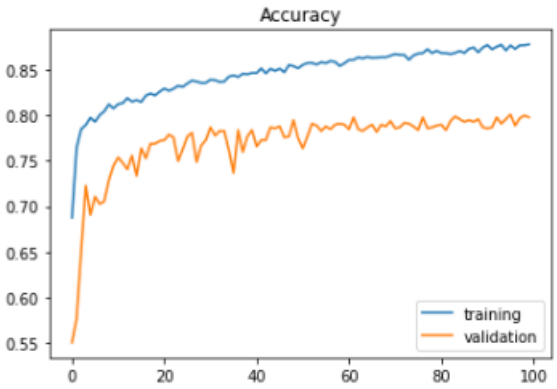
Metric	Score
loss	0.656663715839386
auc	0.8930051922798157
false_negative	179.0
false_positives	179.0
precision	0.8208208084106445
recall	0.8208208084106445
accuracy	0.8208208084106445

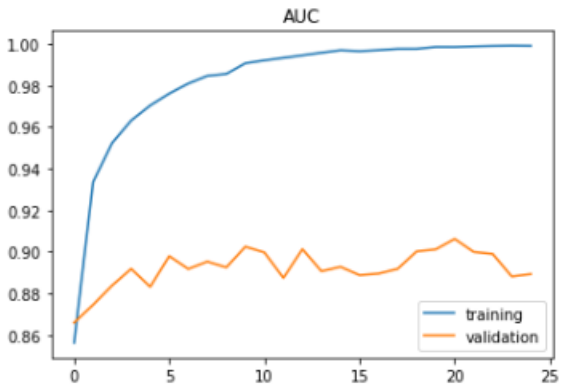


Test Data

Metric	Score
loss	0.4733511507511139
auc	0.9354085922241211
false_negative	80.0
false_positives	80.0
precision	0.8787879347801208
recall	0.8787879347801208
accuracy	0.8787879347801208



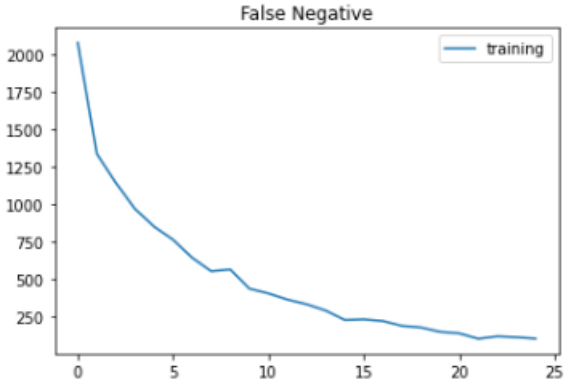


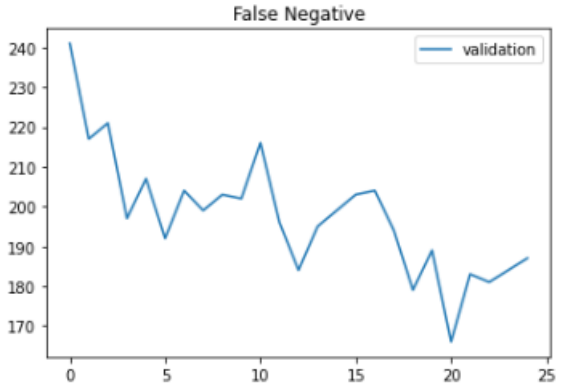


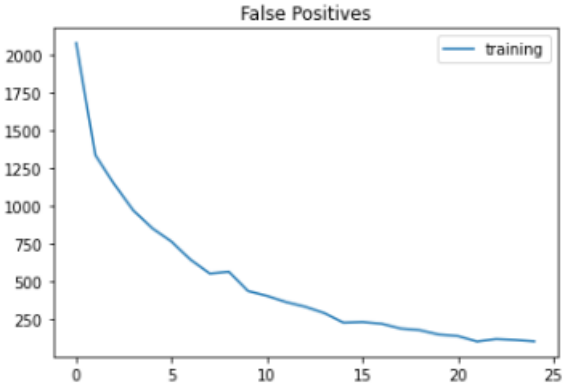
VIII. TRANSFER LEARNING RESULTS

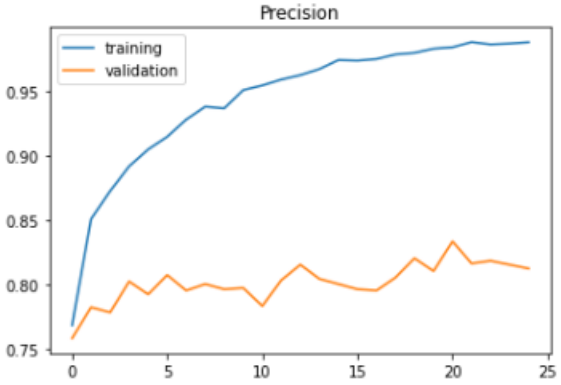
Train Data

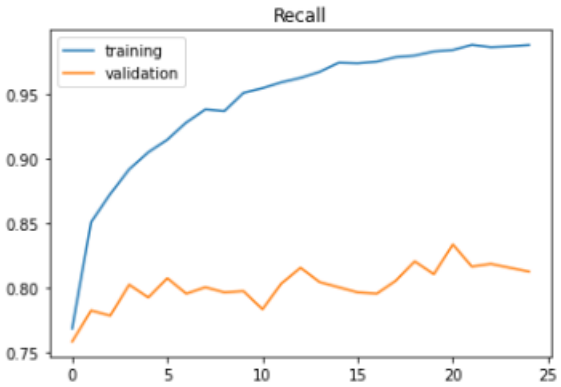
Metric	Score

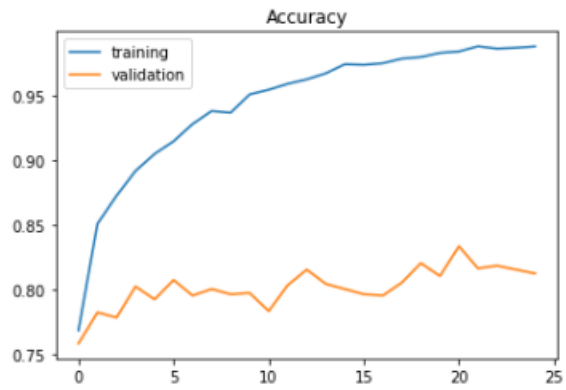












Thus, precision, recall and accuracy are equal for all the datasets respectively.

IX. CONCLUSION

Non-dermoscopic images taken by digital cameras have been serving as a tool for melanoma detection in telemedicine. In this research a computational complex method based on deep learning was implemented that used clinical images. This system was capable of detecting melanoma cases from benign ones. We were able to increase the accuracy of the system by sending images through illumination correction that increased the discrimination capability of the system. For training, we used an available small dataset. By cropping, scaling, and rotating of images the number of images was increased. Our proposed method left the process of feature extraction to CNN while traditional learning approaches try to extract features from data. Experimental results showed our better accuracy, as compared to other detection algorithms.

REFERENCES

[1] G E. Khvedchenya V. I. Iglovikov A. Buslaev, A. Parinov and A. A. Kalinin. Albumentations: fast and flexible image augmentations. ArXiv e-prints, 2018.

[2] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. arXiv preprint arXiv:1708.04552, 2017.

[3] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 7132–7141, 2018.

[4] Mingxing Tan and Quoc V Le. Efficientnet: Rethinking model scaling for convolutional neural networks. arXiv preprint arXiv:1905.11946, 2019.

[5] Skincancer.org, "Melanoma - SkinCancer.org," 2016. [Online]. Available: http://www.skincancer.org/skin-cancer-information / .

[6] J. F. Alcon, C. Ciuhu, W. Ten Kate, A. Heinrich, N. Uzunbajakava, G. Krekels,D.SiemandG.DeHaan,"Automaticimaging system with decision support for inspection of pigmented skin lesions and melanoma diagnosis," IEEE Journal of Selected Topics in Signal Processing, vol.3, no. 1, pp.14-25, 2009.

[7] C.MunteanuandS.Cooclea,"Spotmole– melanoma control system," 2009. Available: http://www.spotmole.com/.

[8] J.Shuiwang,X.Wei,Y.Ming,andY.Kai,"3DConvolutionalneural networks for human action recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 1, pp. 221- 231, 2013.

[9] L. Siqi, L. Sidong, C. Weidong, S. Pujol, R. Kikinis, and D. Feng, "EarlydiagnosisofAlzheimer'sdiseasewithdeeplearning,"inIEEE International Symposium on Biomedical Imaging (ISBI), 2014, pp. 1015-1018, 2014.