Machine Learning Classification on OCT Images from OLIVES Dataset

Stella Fournier and John Grilo
Code: https://github.com/jgrilo3/8803_Final_Project

I. Introduction

For our Fundamentals of Machine Learning final project, we used four different supervised machine learning algorithms to conduct Diabetic Retinopathy Severity Scale (DRSS) severity classification on the Optical Coherence Tomography (OCT) images from OLIVES dataset[4]. Our group chose to perform evaluation on OCT images using K-nearest neighbor (KNN), Naïve Bayes, convolutional neural network (CNN) with an untrained AlexNet, and transfer learning CNN using a pretrained ResNet.

II. Methoedology

A. KNN

i. Method

From the PRIME dataset, we flattened each image pixel value into a 1D array. This allows the images to be represented as a vector of pixel features that can be fed into the KNN algorithm. For this section we focused on the regular KNN algorithm alongside the use of Principial Component Analysis (PCA) on our training set as a dimensionality reduction technique.

ii. Experiment & Analysis

KNN is the simplest model out of the four ML models we employed. The model learns the entire training set and determines the output prediction as the category with the highest number of occurrences among the 'k' closest neighbors according to some distance metric. There are two hyperparameters that we can tweak for KNN, the distance metric and the number of ‘k’.

For our first analysis, we varied the value of 'k' from 1 to 20 by running the flattened image dataset through an iterative loop. We found that the best ‘k’ value is 15 yielding a balanced accuracy of 33.91% which is roughly equivalent to randomly guessing (33.3%). Though looking at Fig. 1, there isn’t any relationship between the hyperparameter ‘k’ and the balanced accuracy likely due to the high number of features present.

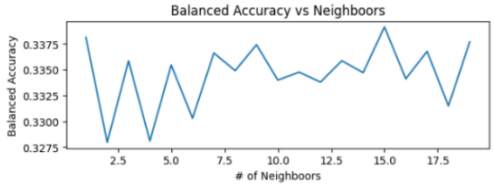


Fig. 1: The number of ‘k’ neighbors and its corresponding balanced accuracy using KNN.

One of the reasons why the balanced accuracy is so low is because KNN suffers from the “curse of dimensionality.” To mitigate this issue, we applied PCA to the original flattened image dataset as a dimensionality reduction technique to extract the most important features based on cumulative variance within the vector. By choosing only the most important features, i.e. features with the largest variance associated with them, the data will hold similar information but in a reduced dimensional space.

To determine the optimal number of components ‘n’ for PCA the optimal value of k was initially considered for the non-PCA affected data. The value of n was increased from 10 to 300 in increments of 10. Fig. 2 shows the results of this process. The optimal number of components peaked early at ~20 components and quickly dropped off by 50 components. This is likely due to redundant features in the images such as large black background or noise outside the main image subject accounting for little variance.

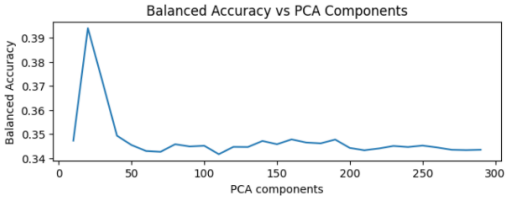


Fig. 2: The number of principal components and their corresponding balanced accuracy with k value equals 15 on training data using KNN.

Further evaluations of the KNN model were performed to see how the sample of n = 10, 20, 50, and 100 pair with k values of 1 to 30 would affect the balanced accuracy. Based on the results in Fig. 3, there is a decaying trend for all different n values when the number of k neighbors increases. Additionally, we can see that PCA with n = 20

outperformed the other n values and k value of 3 has the highest balanced accuracy of 40.93%.

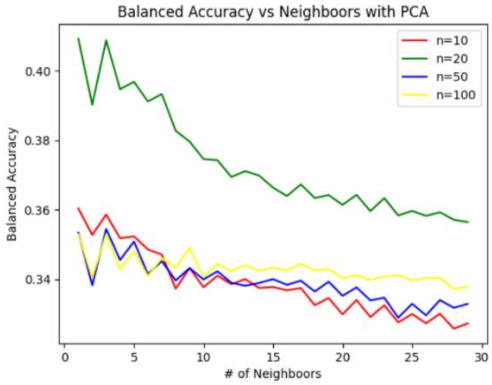


Fig. 3: The number of 'k' neighbors and its corresponding balanced accuracy with n PCA components on training data using KNN.

The balanced accuracy is improved by 20.7% with the implementation of KNN with PCA, compared to using KNN without applying PCA making PCA a valuable technique for this dataset when approaching it using KNN.

B. Naïve Bayes

i. Method

Naïve Bayes was the second model used for classification of the OCT images. The data was prepared in an identical way to KNN: the images were flattened into 1D arrays containing the entirety of the image pixels. Initially, testing consisted of running the flattened data to achieve a baseline balanced accuracy. After that, tests using PCA were conducted to find potential improvements in balanced accuracy over the latter.

ii. Experiment & Analysis

Historically, Naïve Bayes is not an ideal model for image data classification as it assumes independence between features while image pixels are highly correlated.

Running the dataset using Naïve Bayes, we can already see an improvement in the balanced accuracy from KNN with a balanced accuracy of 46.84%. To further improve the accuracy, we applied PCA with n from 10 to 300 with an increment of 10 then applied inverse transform on the training data to bring it to the size of the original image. As shown in Fig. 4, a PCA component value of 30 has the highest balanced accuracy of 51.22% which is a 20.09% increase over the highest balanced accuracy obtained from KNN. The likely reason for this is the inner works of Naïve Bayes. Since the Naïve Bayes algorithm assumes independence, it functions better on high dimensionality data. However, with such high amounts of equally important features, the algorithm will have to consider redundant pixels thus leading to poor classification. Once PCA modifies the data, the high variance components are preserved thus reducing redundancies and leading to a more accurate Naïve Bayes model.

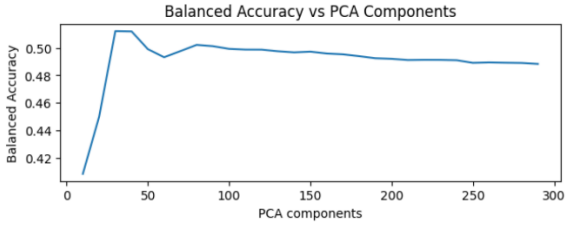
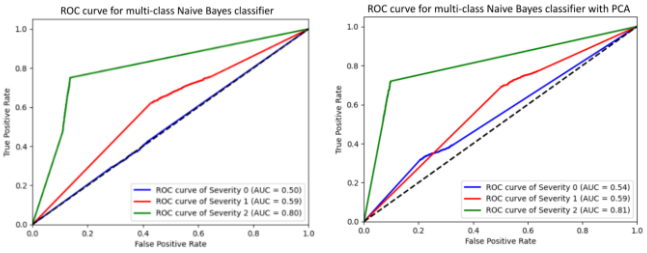


Fig. 4: Balanced accuracies at different numbers of PCA components for Naïve Bayes.

Comparing the AUC values from the ROC curves in Fig. 5, it appears that Naïve Bayes classifier with PCA has a slightly better discrimination ability for severity level 0 and severity level 2, whereas it performs the same for severity 1. Nevertheless, the difference in AUC values between the two models is small and in both cases severity 0 and severity 1 are close to what is obtained through randomly guessing. Furthermore, upon comparing the performance metrics for both models, Naïve Bayes with PCA has higher recall, precision, and F1 score values for all three classes, indicating better performance overall (Table 1). This makes the Naïve Bayes classifier with PCA performs better than without PCA. This can potentially be attributed to PCA's ability to eliminate irrelevant features, reduce the risk of overfitting, improve signal-to-noise ratio, and increase the separability of the different classes.



(a) (b)

Fig. 5: ROC curves for multi-class Naïve Bayes classifier.

	Naïve Bayes	Naïve Bayes with PCA
Recall	[0.2928, 0.6238, 0.4885]	[0.3179, 0.6931, 0.5313]
Precision	[0.3221, 0.5804, 0.5093]	[0.4173, 0.5690, 0.6349]
F1	[0.3067, 0.6014, 0.4987]	[0.3609, 0.6250, 0.5785]

Table 1. Recall, Precision, and F1 Scores for Naïve Bayes with and without PCA. [Class 0, Class 1, Class 2]

C. CNN with AlexNet

i. Method

The third ML model chosen was using an untrained AlexNet architecture because it is a simpler CNN used for classification of 224x224 RGB images. This method involved an untrained AlexNet,[1] that was trained on

the original dataset with different learning rates (batch size = 64, gamma = 0.001, lr = 0.0001, epochs = 10) to obtain an ideal learning rate (Fig. 8). Next, different augmented forms of the dataset to improve performance given an unbalanced dataset that lacks prominent features between image classes (for individual results see Appendix A). The data augmentations explored in this method were cropped and horizontally flipped versions of the data mixed with the original dataset. Finally, visualization took place to see the activations of the images on the testing set to confirm that the model is making use of the features in an appropriate way.

ii. Experiment & Analysis

The untrained AlexNet yielded a balanced accuracy of 35.89% using original dataset alone. The low accuracy of an untrained AlexNet is due to the randomly initialized weights, and imbalanced dataset causing overfitting. To resolve this issue, the group conducted a baseline tuning of this model where learning rate was selected to increase performance in the model. Fig. 8 shows the learning rate curves peaking around lr = 0.0001 indicating this is a better estimate of the learning rate to use. To further address this issue, data augmentation[3] was applied to increase the size of the dataset and introduce more variation into the training data. In our experiment, we create more images through techniques including horizontal flipping and cropping, to expose the model to more diverse examples of each class, thereby helping it to better generalize to new, unseen images.

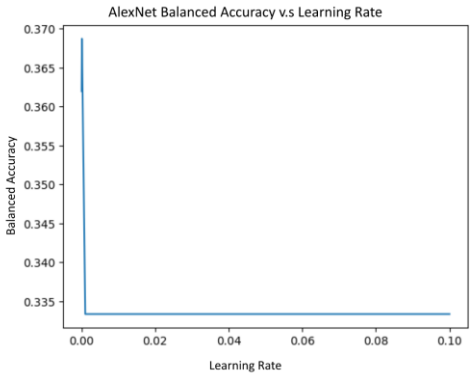


Fig. 5: Balanced Accuracy vs Learning Rate

The first data augmentation technique is cropping (Fig. 7b). The images underwent a center crop, removing 25 pixels rows from the top and bottom resulting in a new image size of (174, 224). Then we add black padding on both the top and the bottom to restore the original image size (224, 224). Given that the layers of the eye are the most critical part of OCT images, cropping the top and bottom of the image eliminates noisy pixels and residual features such as captions in the bottom left corner. We performed the training on the cropped version of the images only which yielded a balance accuracy of 37.48%. Furthermore, a combination of the original dataset and the cropped dataset improved the balanced accuracy to 38.05%. The slight improvement in model performance could be attributed to the larger and more diverse dataset, which allows for better generalization of unseen data.

Horizontal flipping was used to increase the dataset size and make the model more robust against orientation and translations (Fig. 7c). The images were flipped and appended to the original dataset which yielded a balanced accuracy of 35.74%. Additionally, a data set consisting of the original images plus the flipped version of Severity 2 were concatenated. The resulting model produced the highest balanced accuracy of the Alexnet results with a balanced accuracy of 42.98% due to the balancing of class samples to within 3000 samples of each other.

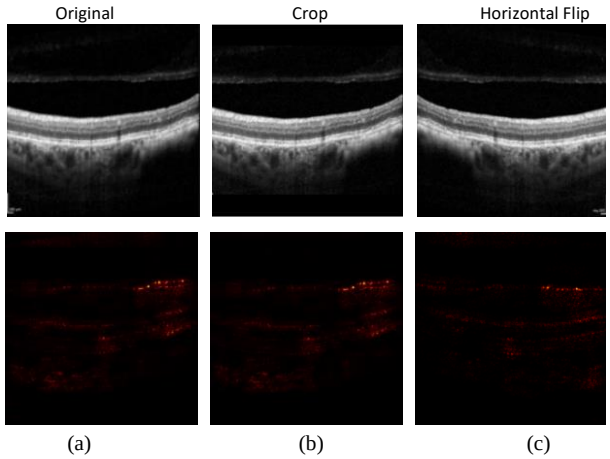


Fig. 6: Original trainset image and undergone data augmentation example and their corresponding visualization.

As a final metric, visualization via saliency mapping[2] was used to show the activations of the features. This helps determine if the model is focusing on the right parts of the image and the reasoning for its decision. Looking at Fig 6, the activation of the Alexnet model is consistent with the white parts of the image indicating the model is focusing on the eye to determine the severity level. One distinction made is within the cropping saliency map that shows a removal of edge feature activations due to the cropping. This seems to center the activation along the eye capture allowing the model to capture that feature in more detail thus explaining the boost in balanced accuracy.

D. CNN with ResNet

i. Method

The last model chosen for the project is transfer learning with the ResNet model. ResNet architecture is based on the idea of residual learning, which adds shortcut connections between layers to help alleviate the vanishing gradient problem. The data is prepared in a similar way to AlexNet (see section C). A pretrained ResNet model was trained on its fully-connected layers using the original dataset with different learning rates (batch size = 64, gamma = 0.001, lr = 0.0001) and data augmentation variants for the training set.

ii. Experiment & Analysis

Running the original dataset to the ResNet model yielded a balanced accuracy of 38.76%. Like AlexNet, data augmentation technique was used to increase the size of dataset and introduce more variation into the training data to increase the accuracy. The first data augmentation technique is cropping. The same methodology is used: cropping 25 pixels from the top and bottom then applying black padding to restore the original image size. The Resnet model on the cropped images yielded a balanced accuracy of 35.75%. Since the accuracy worsened, we applied Resnet model on the cropped images appended to the original images, increasing the data size. This resulted in a better-balanced accuracy of 39.58%

Horizontally flipped images is the second data augmentation technique used. We flipped all images and appended them to our original dataset which yielded us a balanced accuracy of 34.02%. Additionally, we tested another model by only appending the horizontally flipped images in Severity Class 2 to the original dataset due to the lack of data images in Severity Class 2 compared to the other two classes. With this model, we yielded a balanced accuracy of 35.75%.

Data Augmentation	Balanced Accuracy
Original	38.76%
Crop	35.75%
Original + Crop	39.58%
Original + Flip	34.02%
Original + Flip2	35.76%

Table 2. Balanced accuracy for data augmentation for ResNet model.

Looking at the differences in accuracy in Table 2. The balanced accuracies do not have much better performance over the base dataset if any improvement at all. The likely explanation for these results comes from ResNet learning capacity. Since ResNet is a much newer and more complex model than Alexnet, it holds more learning capacity. That would cause the model to start overfitting to noise. This would increase accuracy on the train and validation set as in Fig 7 while sacrificing generalizability on the unseen test data. Fig 8 shows a good representation of this through the saliency map. The ResNet model is activating all over the image rather than just the eye meaning it is attempting to fit the unique noise pattern as the image fades to a back background.

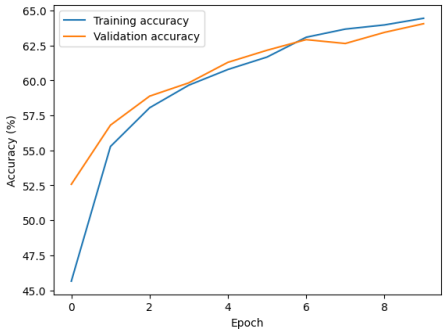


Fig. 7: Train and Validation vs # Epochs

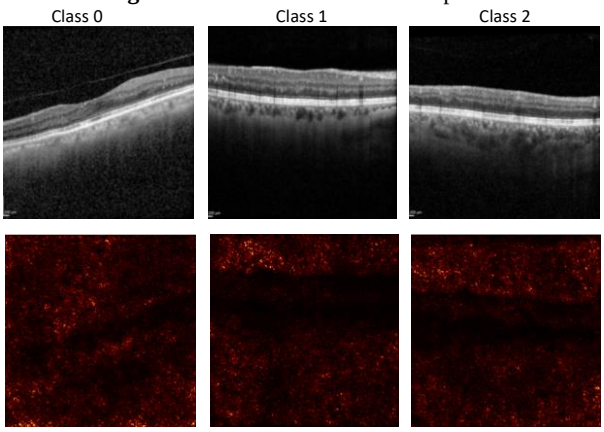


Fig. 8: Original + Flip Severity Class 2 Saliency Mapping

E.) Conclusion

Out of the four ML models performed on the OCT images, Naïve Bayes with PCA had the overall highest balanced accuracy of 51.22%. This could be because Naïve Bayes is a relatively simple and fast algorithm that works well with high-dimensional data, such as the OCT images used in this project. Additionally, applying PCA to reduce the dimension of the data makes it easier for the Naive Bayes algorithm to classify the images.

For the deep learning models, ResNet yielded a highest accuracy of 42.98% for horizontally flipped images in class two appended to original dataset. AlexNet yielded a highest accuracy of 43.51% on the original dataset. These two deep learning models did not perform as well as Naïve Bayes likely because they require large amounts of data and computational resources to train effectively. For our dataset, it's possible that the limited size of dataset and the complexity of the OCT images made it difficult for the deep learning models to learn meaningful representations of the images.

References

[1] X. Wang, Y. Lu, Y. Wang and W. -B. Chen, "Diabetic Retinopathy Stage Classification Using Convolutional Neural Networks," 2018 IEEE International Conference on Information Reuse and Integration (IRI), Salt Lake City, UT, USA, 2018, pp. 465-471, doi: 10.1109/IRI.2018.00074.

[2] Nevarekar, Sunny. (2019). pytorch-saliency-maps: Generating Saliency Maps in PyTorch [Source code]. https://github.com/sunnynevarekar/pytorch-saliency-maps

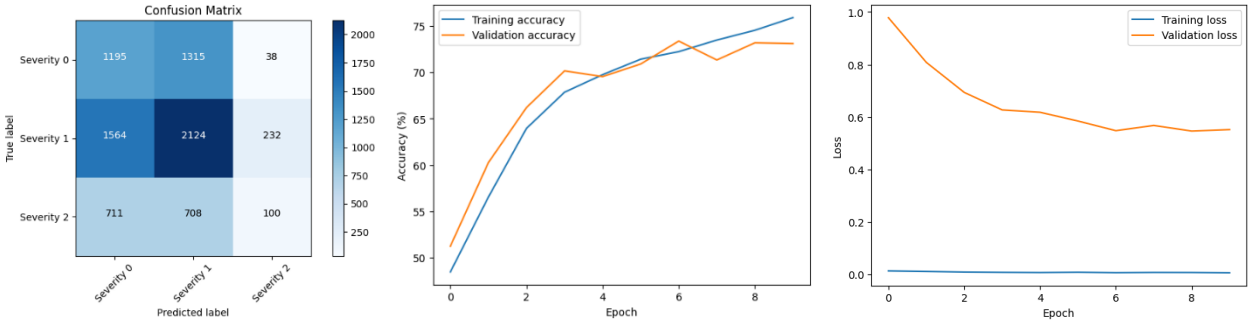
[3] Huang, H., Liu, X., & Zhang, C. (2019). A survey of deep learning for scientific data processing. Journal of Big Data, 6(1), 79. https://doi.org/10.1186/s40537-019-0197-0

[4] M. Prabhushankar, K. Kokilepersaud*, Y. Logan*, S. Trejo Corona*, G. AlRegib, C. Wykoff, "OLIVES Dataset: Ophthalmic Labels for Investigating Visual Eye Semantics," in Advances in Neural Information Processing Systems (NeurIPS 2022) Track on Datasets and Benchmarks, New Orleans, LA, Nov. 29 - Dec. 1 2022

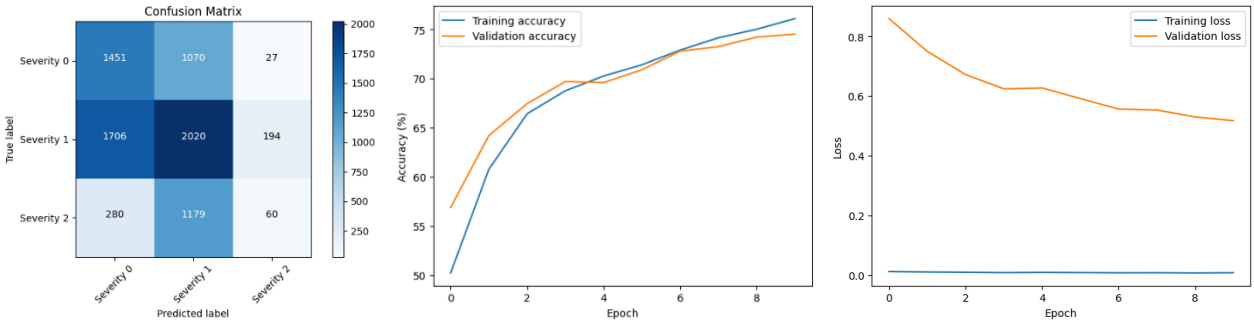
Appendix

AlexNet:

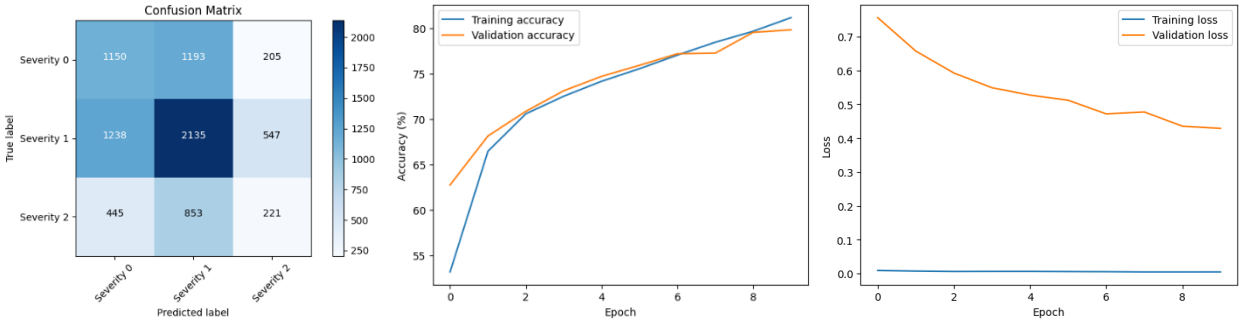
Original



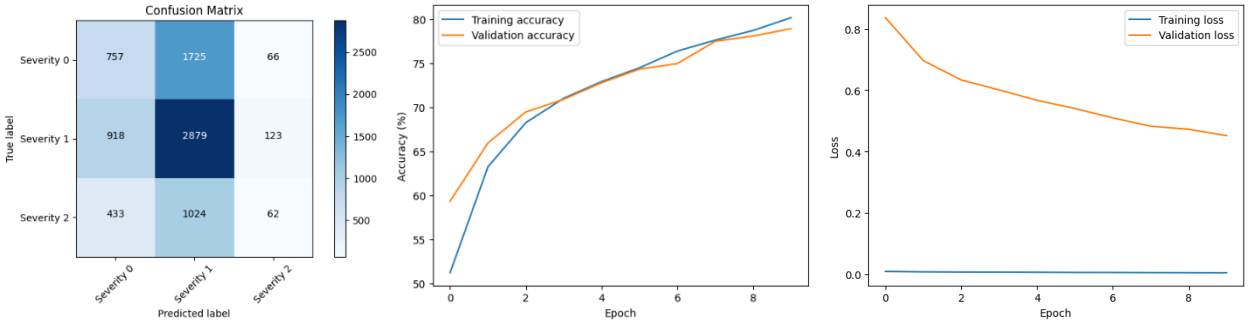
Crop



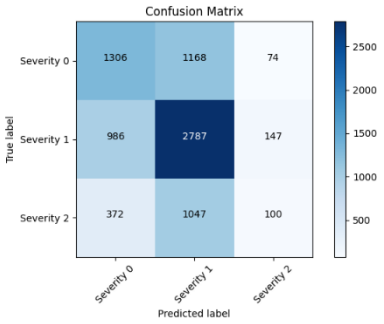
Original + Crop

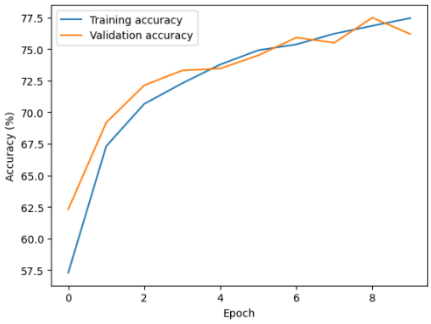


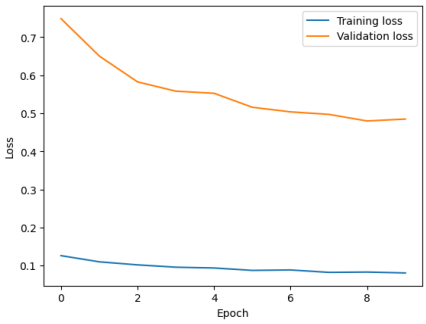
Original + Flip



Original + Flip Class 2

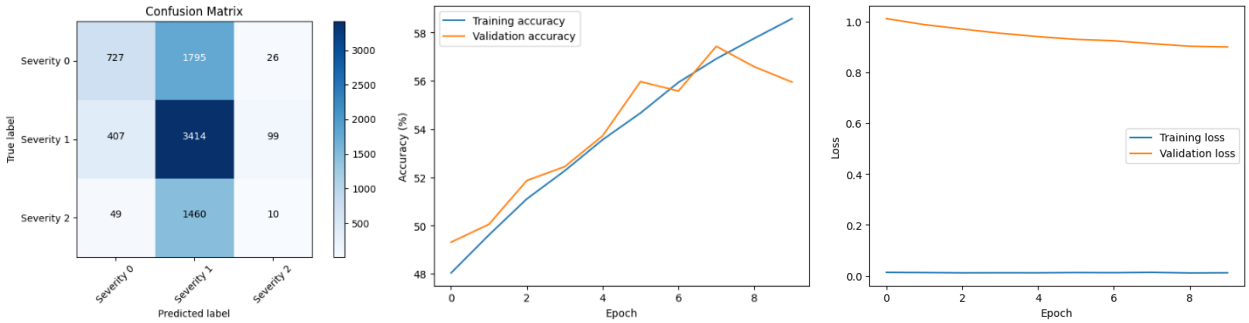

Confusion Matrix
Severity 0
Severity 1
Severity 2
True label
1306
1168
74
986
2787
147
372
1047
100
Severity 0
Severity 1
Severity 2
Predicted label
2500
2000
1500
1000
500


Training accuracy
Validation accuracy
Accuracy (%)
77.5
75.0
72.5
70.0
67.5
65.0
62.5
60.0
57.5
0
2
4
6
8
Epoch

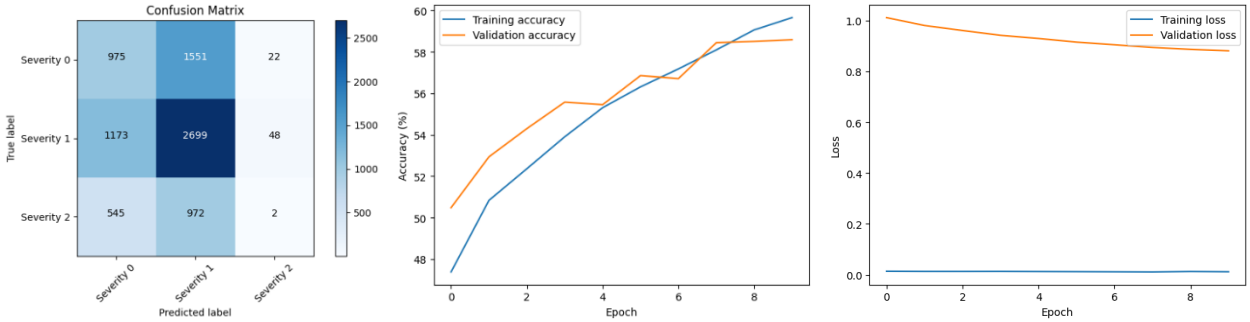

Training loss
Validation loss
Loss
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
2
4
6
8
Epoch

ResNet

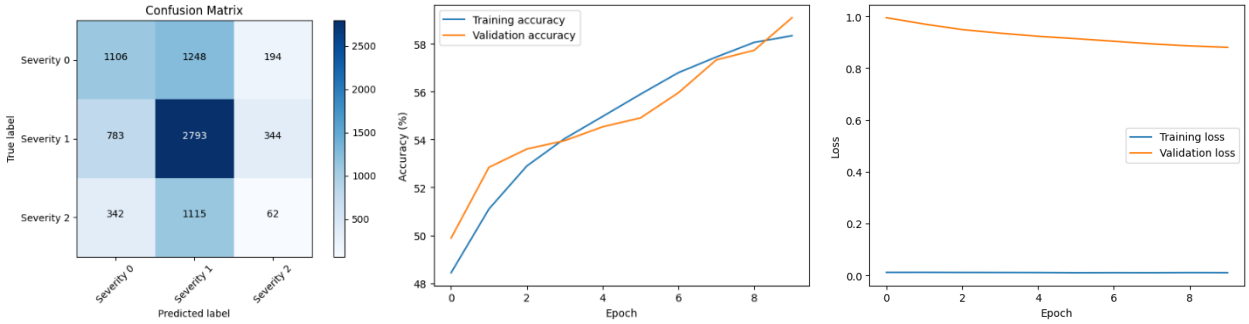
Original



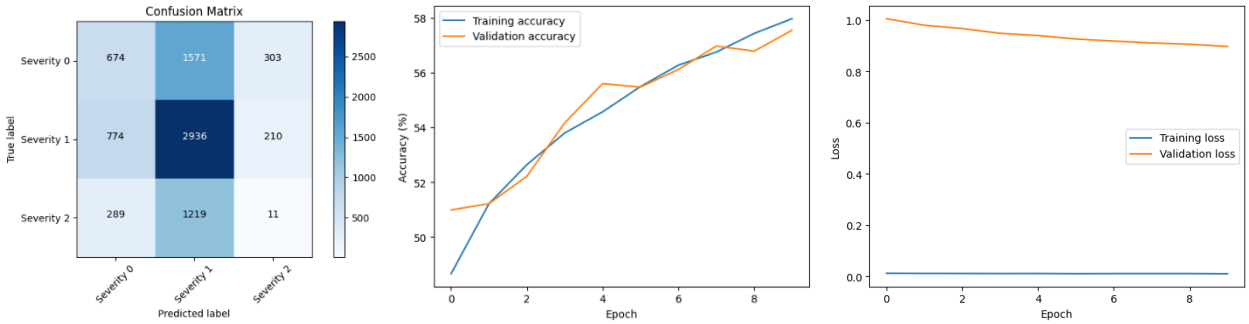
Crop



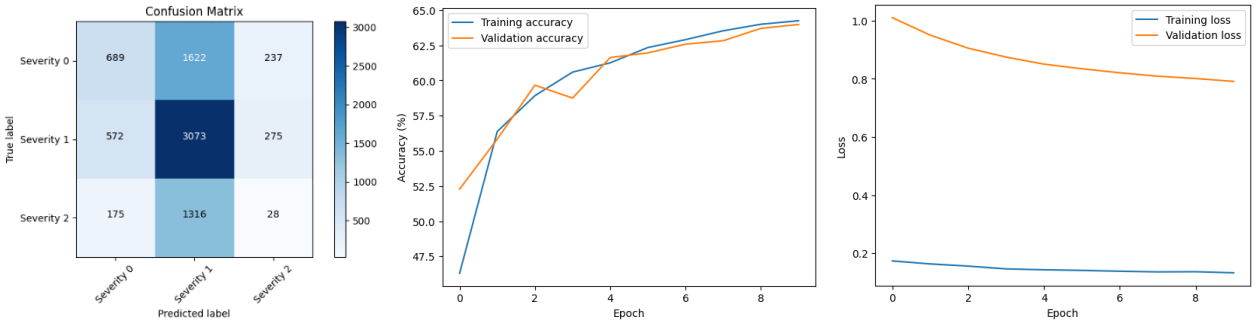
Original + Crop



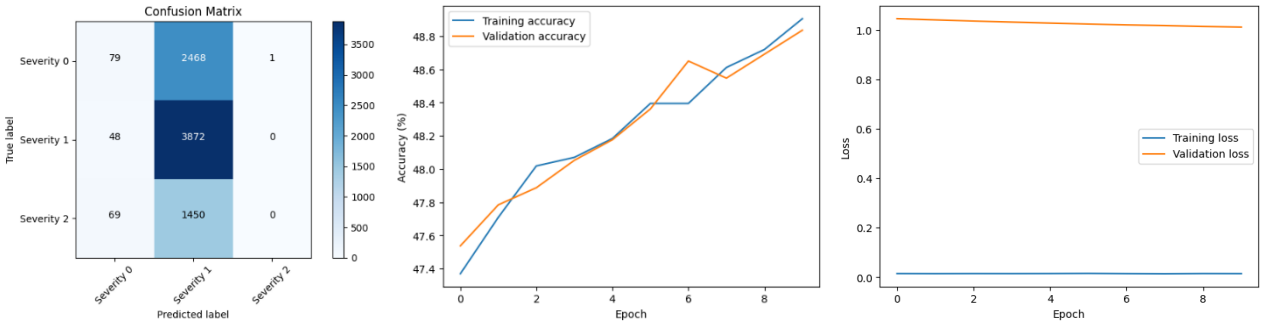
Original + Flip



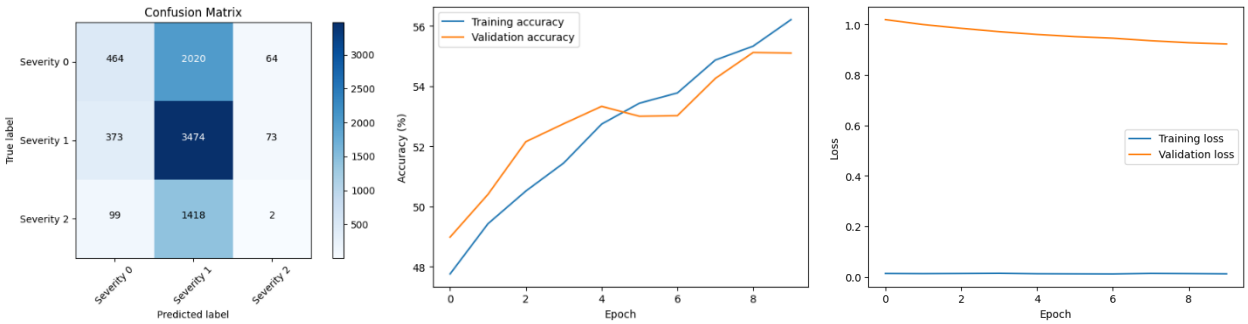
Original + Flip 2



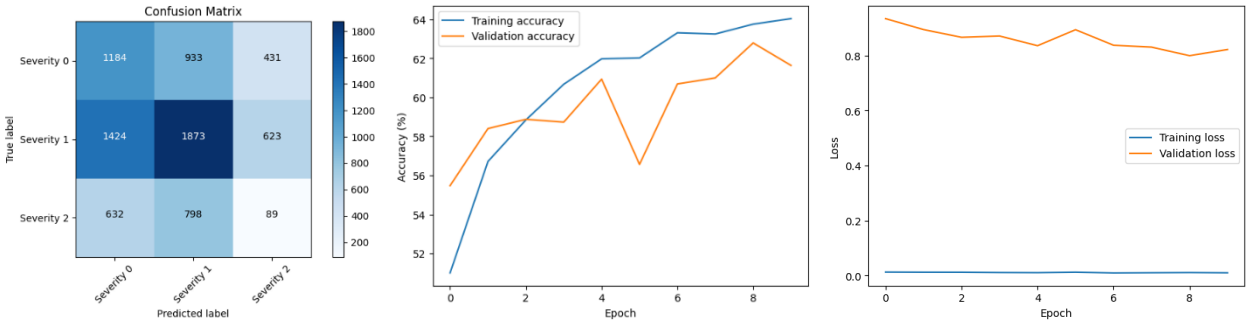
Lr = 0.00001



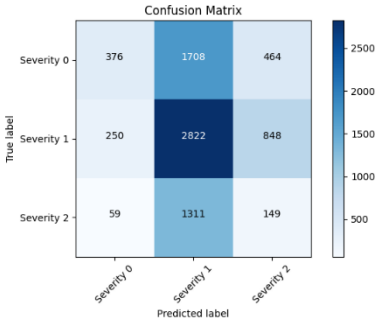
Lr = 0.0001

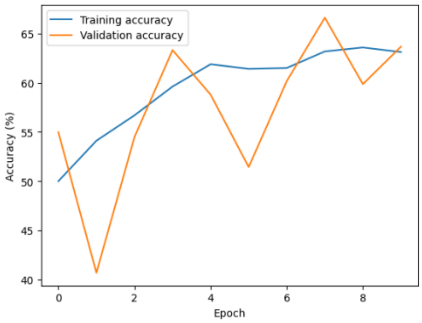


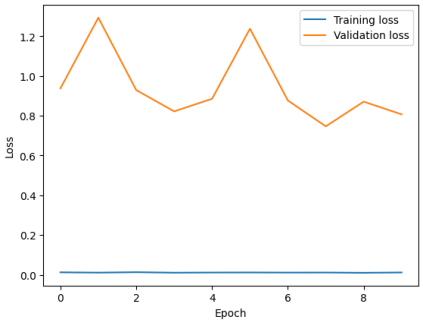
Lr = 0.001



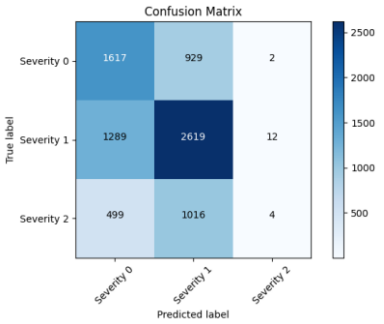
Lr = 0.01

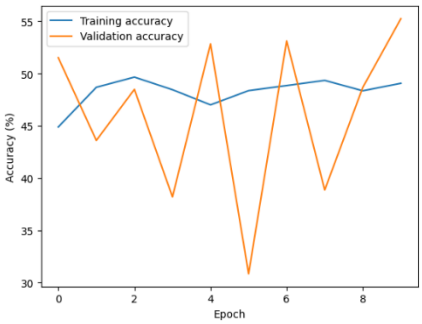

Confusion Matrix
Severity 0
Severity 1
Severity 2
True label
376
1708
464
250
2822
848
59
1311
149
Severity 0
Severity 1
Severity 2
Predicted label
2500
2000
1500
1000
500

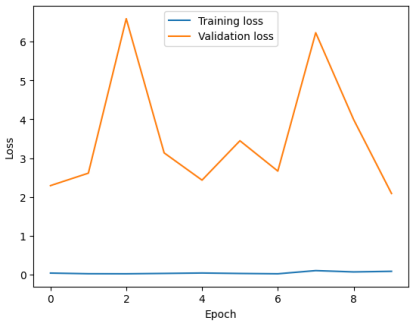

Training accuracy
Validation accuracy
Accuracy (%)
Epoch


Training loss
Validation loss
Loss
Epoch

Lr = 0.1


Confusion Matrix
Severity 0
Severity 1
Severity 2
True label
1617
929
2
1289
2619
12
499
1016
4
Severity 0
Severity 1
Severity 2
Predicted label
2500
2000
1500
1000
500


Training accuracy
Validation accuracy
Accuracy (%)
Epoch


Training loss
Validation loss
Loss
Epoch