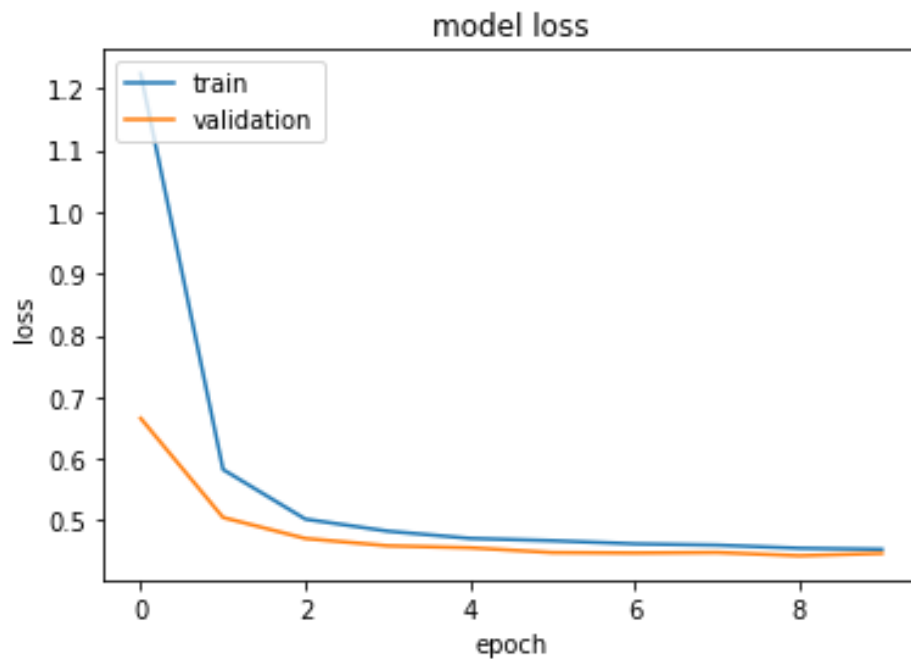
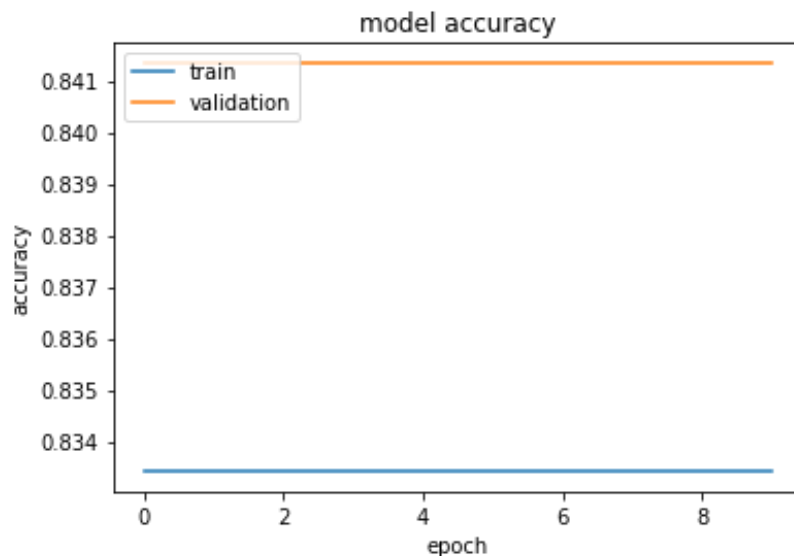


1. Final Training Results

Since my preliminary results were not satisfactory, I continued trying with my CNN's hyperparameters to get more consistent results. I changed the output layer's activation to sigmoid since it is better for binary classification and added L2 regularization in both hidden layers since my model was wildly overfitting. With regularization added, the loss constantly decreases as shown in the figure below.



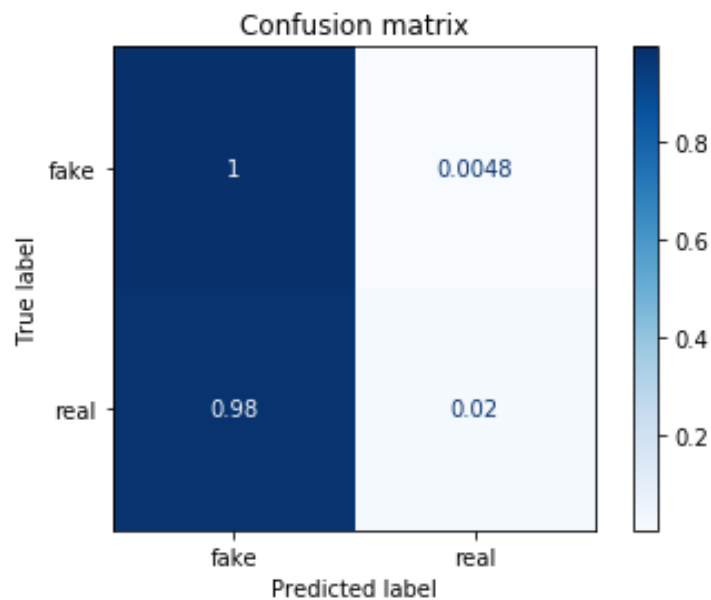
However, the accuracy itself does not change significantly:



I also tried to implement a Random Forest Classifier model for which I obtained the follow accuracies:

```
Training accuracy: 1.0  
Testing accuracy: 0.83
```

And plotted the Confusion Matrix:



From the matrix, we can see the model is not appropriate as it seems to constantly guess false, thus achieving an accuracy of 83%, which is roughly the percentage of fake pictures. Thus, the CNN yields better results and will be used in the fair.

2. Final demonstration proposal.

For the final product, I will create a webapp to demo my project in action. I will let the user choose an online picture that is either a normal face or a deepfake picture, and let my model predict the class. I will use Flask for the backend, since the application is relatively simple and it allows me to import my pre-trained weights that I saved. For the frontend, a simple landing page using HTML and CSS will do. I have previous experience working with Flask, HTML and CSS at previous hackathons, and will sharpen my knowledge of them with YouTube videos.