Instead of random forest, A deep learning model is used in this week’s project.

In specific, A CNN is used. Convolutional Neural Networks (CNNs) are often considered better than Random Forests for specific types of tasks, particularly those involving high-dimensional data like images, videos, and spatial-temporal datasets.

CNNs automatically detect and learn the important features from raw data during the training process, thanks to their convolutional layers. This capacity for automatic feature learning is particularly effective for complex, high-dimensional data like images, where manually designing features is impractical.

Also, They are designed to scale well with high-dimensional data, making them suitable for tasks involving large images or videos. CNNs can manage this complexity without a significant increase in required computational resources relative to the size of the data.

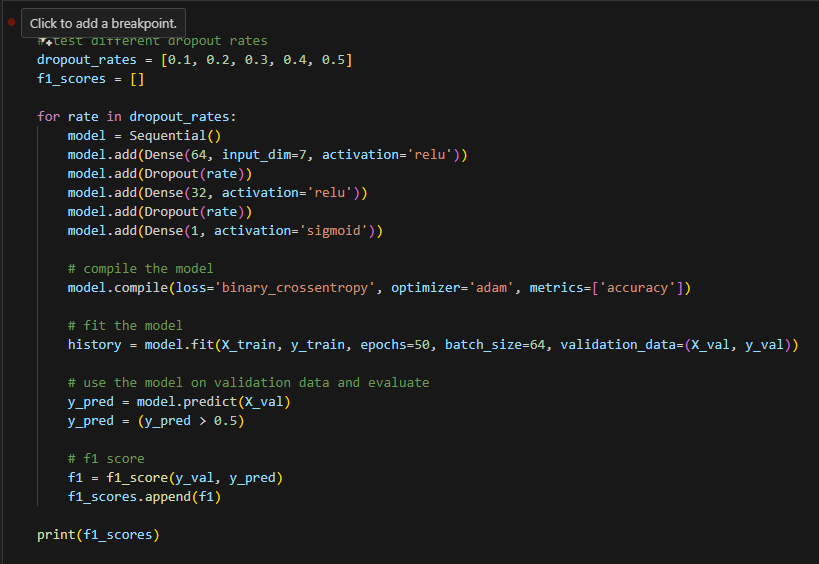
Due to their architecture, CNNs are better at generalizing from the training data to new, unseen data, especially when it comes to images. This is because they learn abstract features that are more universally applicable across different visual domains. Random forest can be more likely to overfit to a dataset.

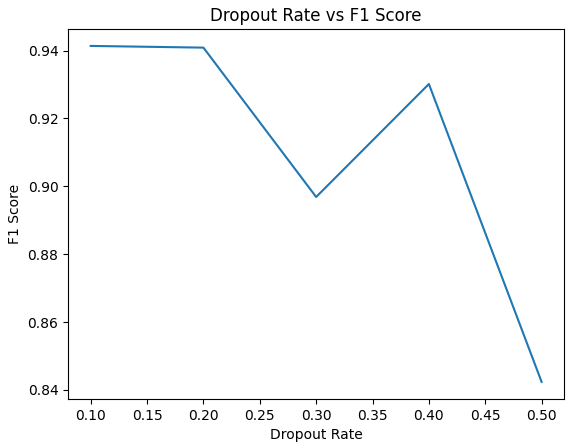
Different Setting is tested for this project. Epochs do not influence this model a lot due to it is rather a simple problem for the CNN model.

Dropout is a regularization technique used in the training of neural networks, including CNN, to prevent overfitting. The concept of dropout involves randomly "dropping out" a proportion of neurons (i.e., setting their activations to zero) in a layer during each training iteration. By "dropout rate," we refer to the probability that each neuron in the layer is dropped. For example, a dropout rate of 0.5 means there's a 50% chance that any given neuron will be set to zero during a particular forward or backward pass.

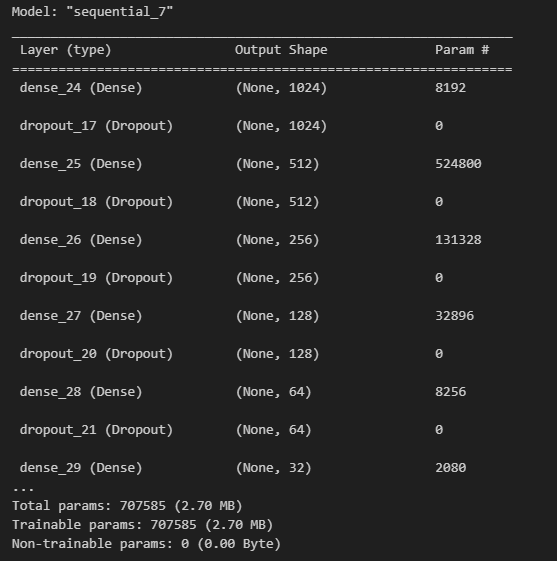
Dropout is not equally effective at all locations within a CNN. It's most commonly applied after fully connected layers or dense layers towards the end of the network. Applying dropout after convolutional layers is also possible, but it's typically done with a lower dropout rate, considering the different nature of spatial feature extraction in these layers.

I Tested 5 different dropout rates, from 0.1 to 0.5.





More Dropout rate seems to require higher epochs so 0.15 is chosen for this project. Here is the final look of the model itself:



And here is the result of this model: