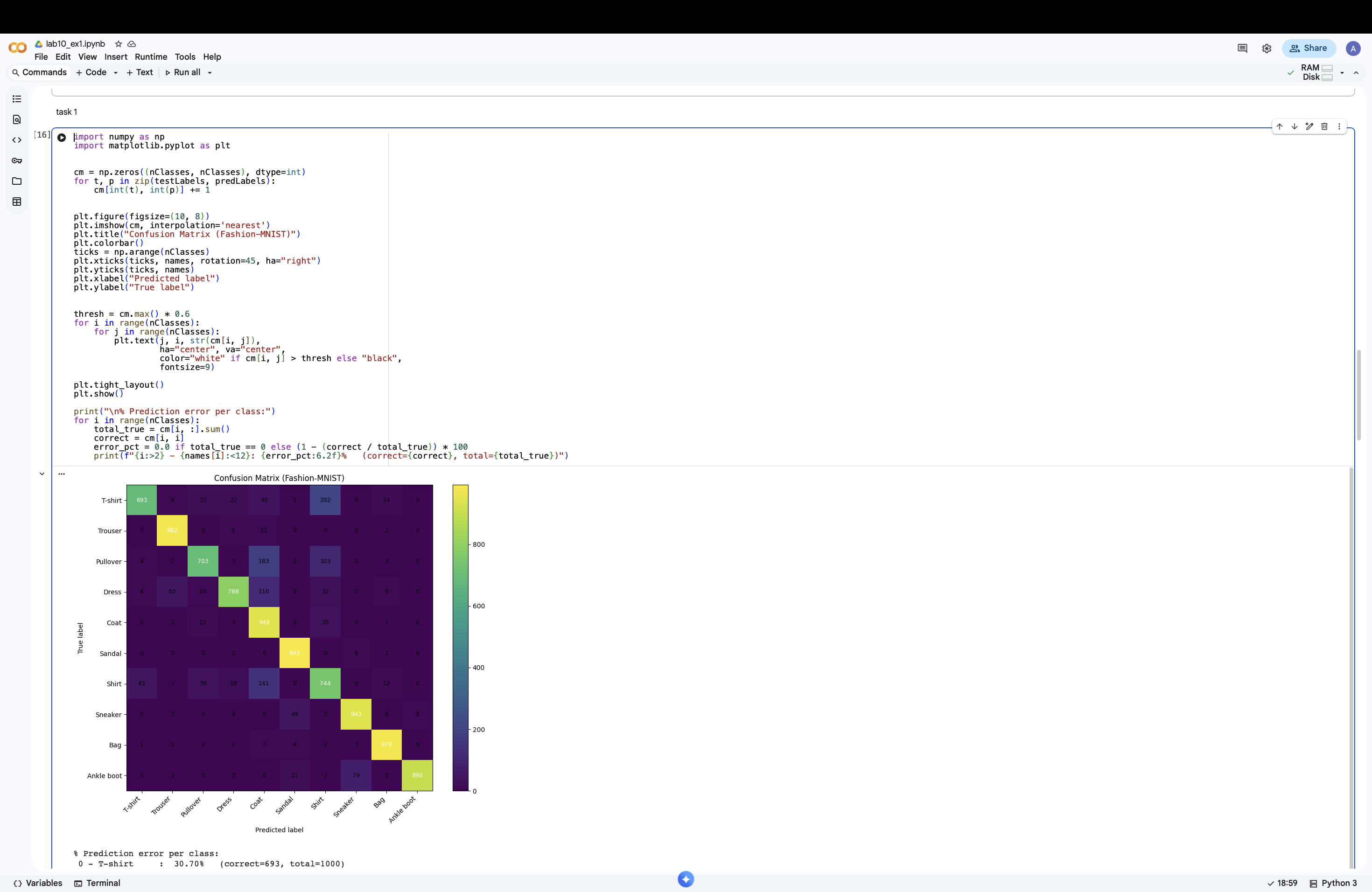
## LAB 10 – CNN

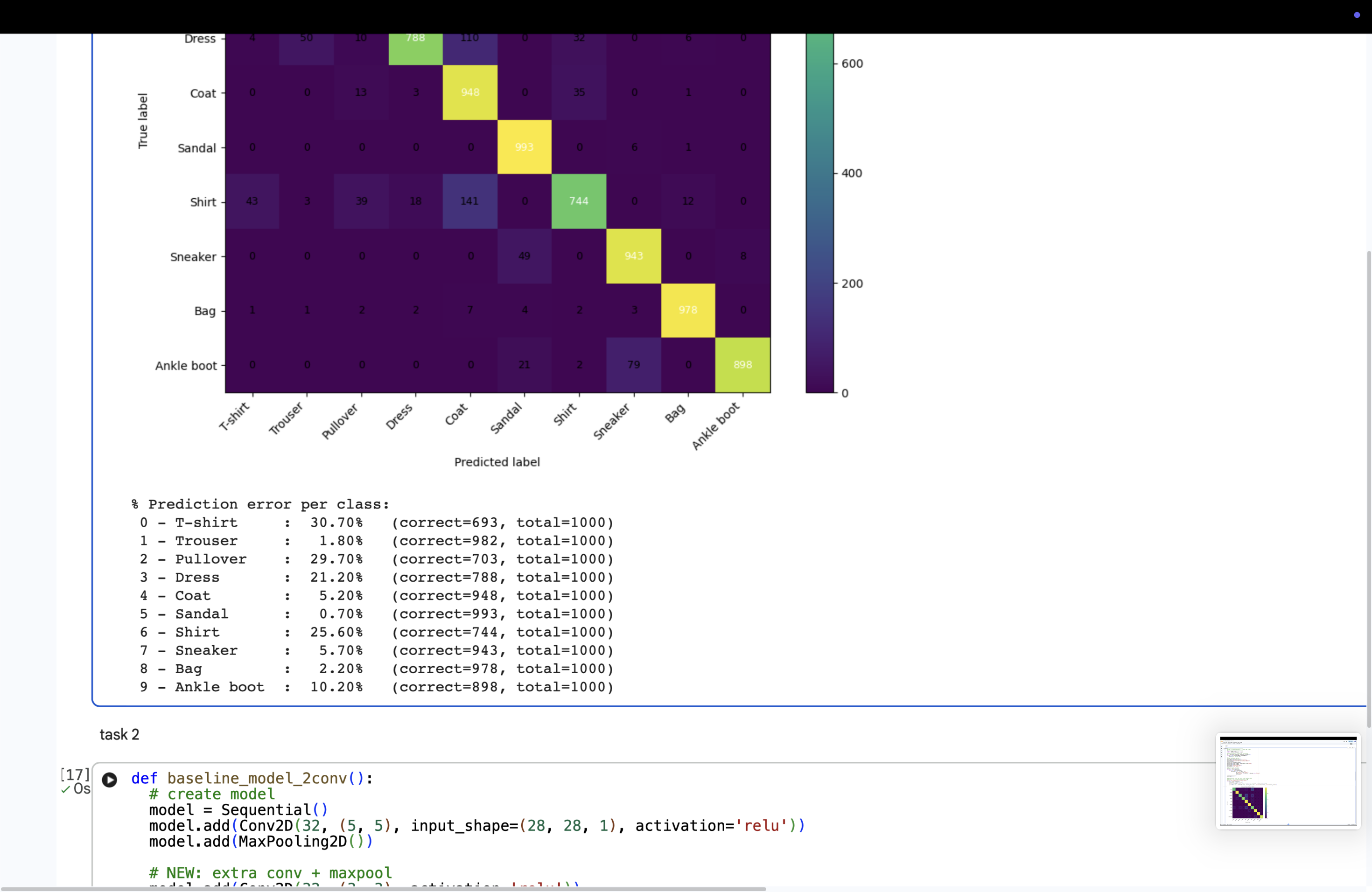
**Bashar Shoumali: 318699154**

**Ameer Rhaiby: 213932718**

## Step 1 – Confusion Matrix

After training the model, a confusion matrix was created to analyze classification results for each clothing class. For every class, the error percentage was calculated using: error% = (total samples – correct predictions) / total samples × 100. This helped identify which classes (like Shirt or Coat) had the highest misclassification rates.

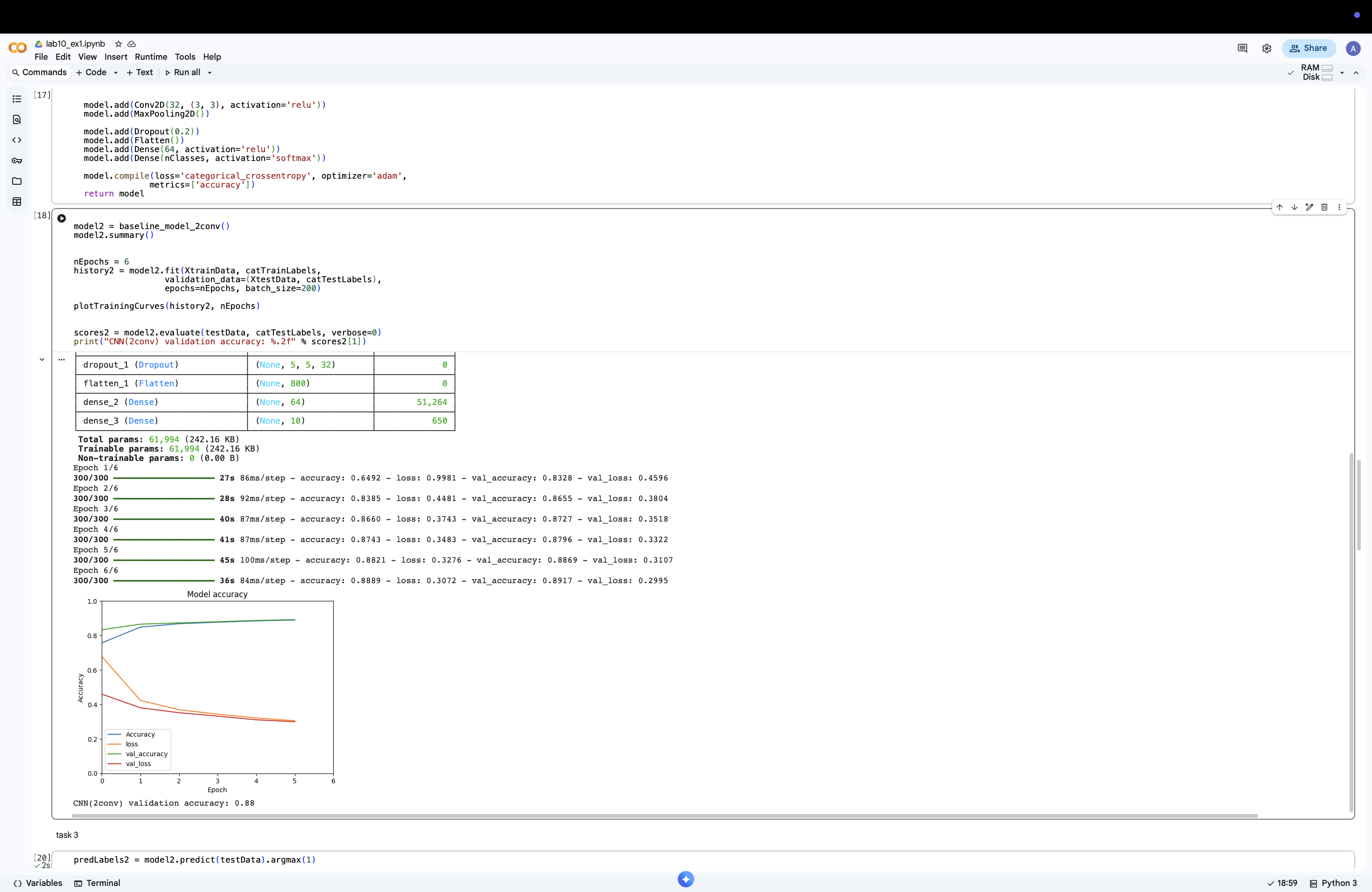
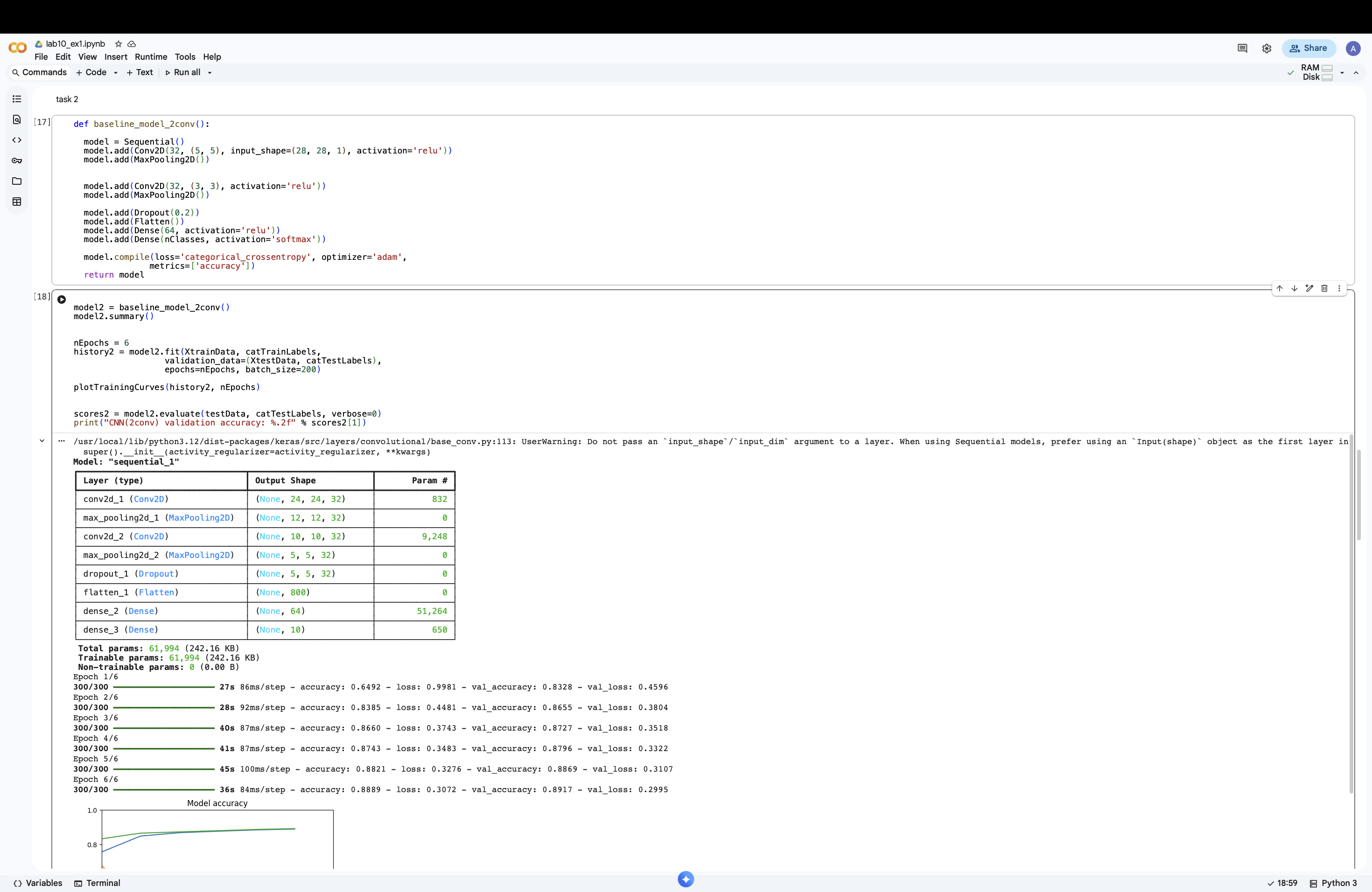




## Step 2 –

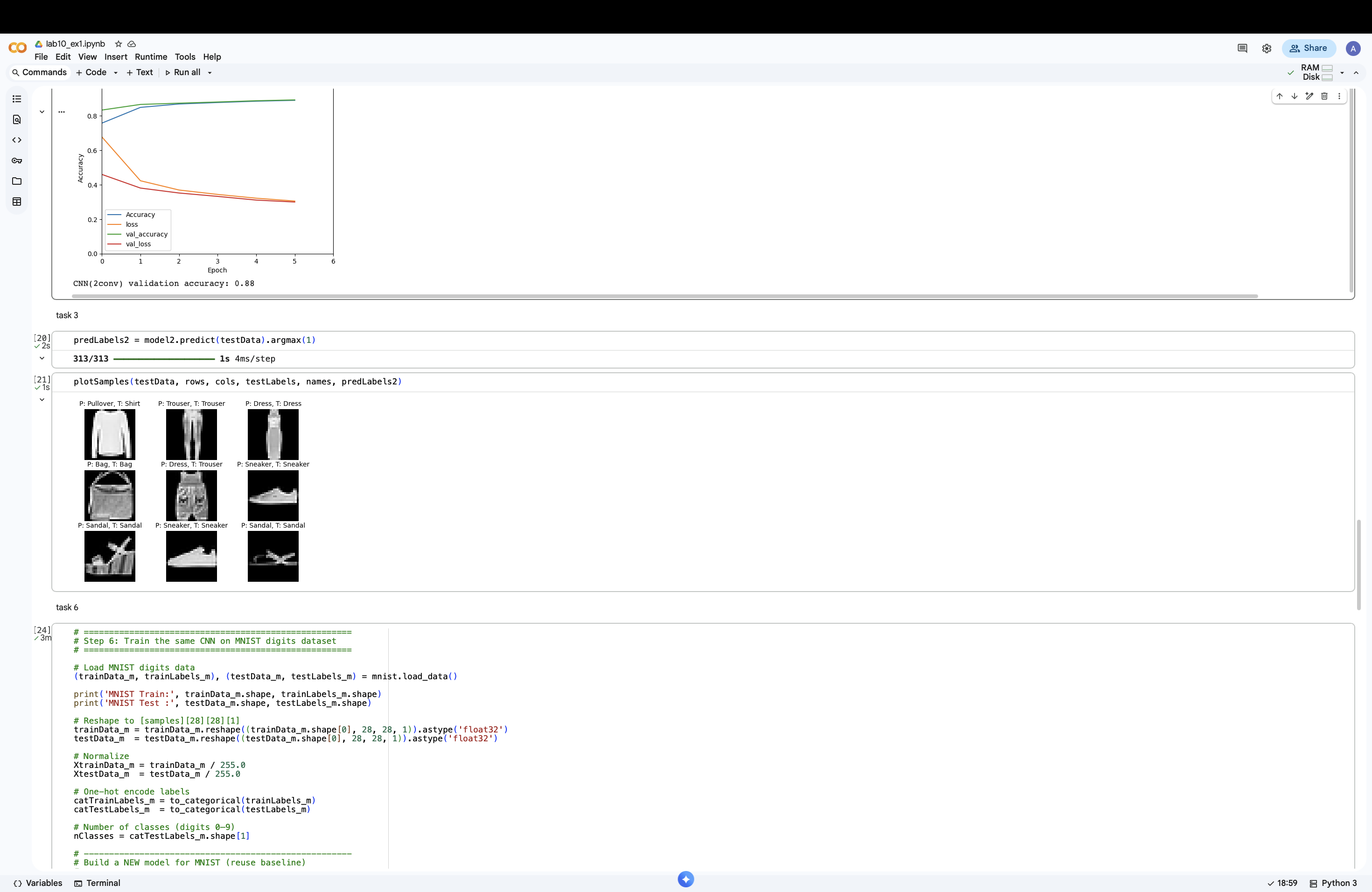
## Adding a Second Convolution Layer

Made the same model with a second convolutional layer with 32 filters of size 3×3.



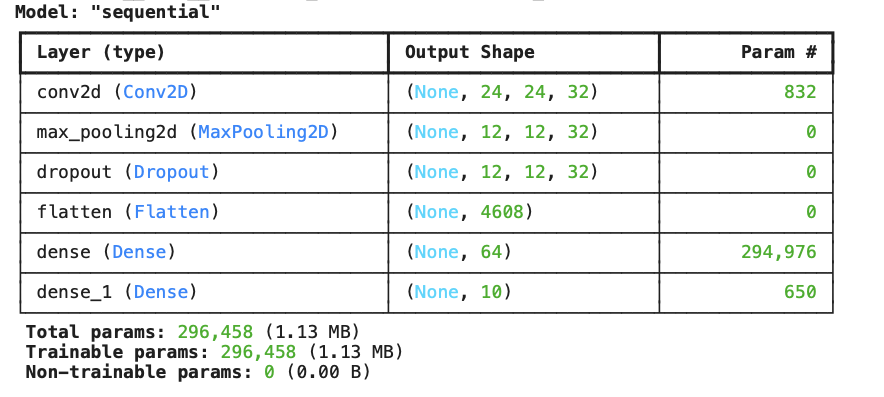
## Step 3 – Visualizing Predictions

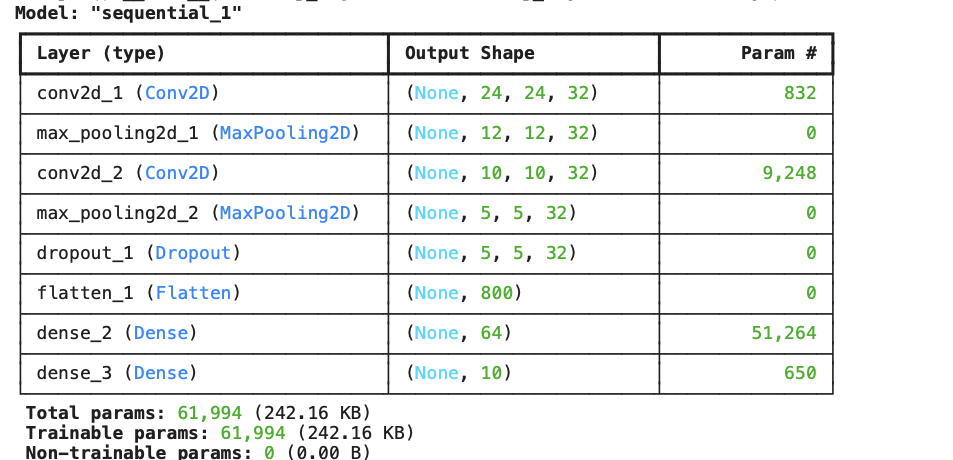
Nine random test images were displayed together with their true and predicted labels.



## Step 4 – Comparing Number of Parameters and Runtime

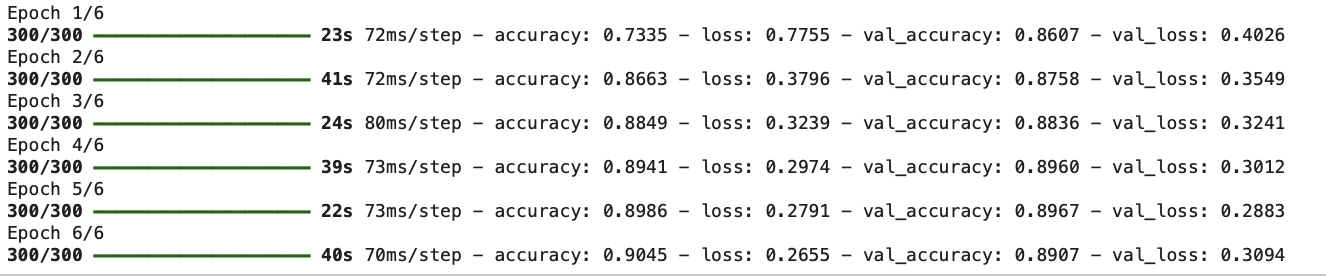
The original model has 296,458 parameters, while the enhanced model has only 61,994 parameters. This happens because after adding the second convolution layer, the feature map size is reduced before reaching the dense layer (from 4608 to 800), which reduces the dense layer parameters: 4608×64+64 = 294,976 vs. 800×64+64 = 51,264. Although the second model has an extra convolution layer, it runs faster and uses less memory because the dense layer dominates the total parameter count.

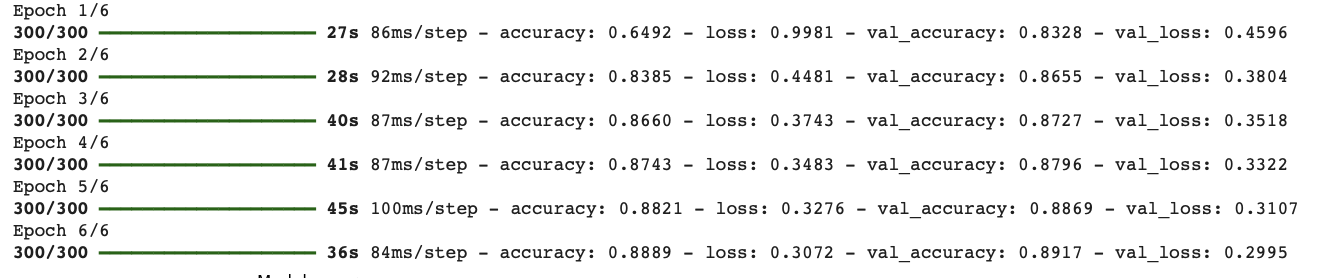




## Step 5 – Validation Curves Comparison

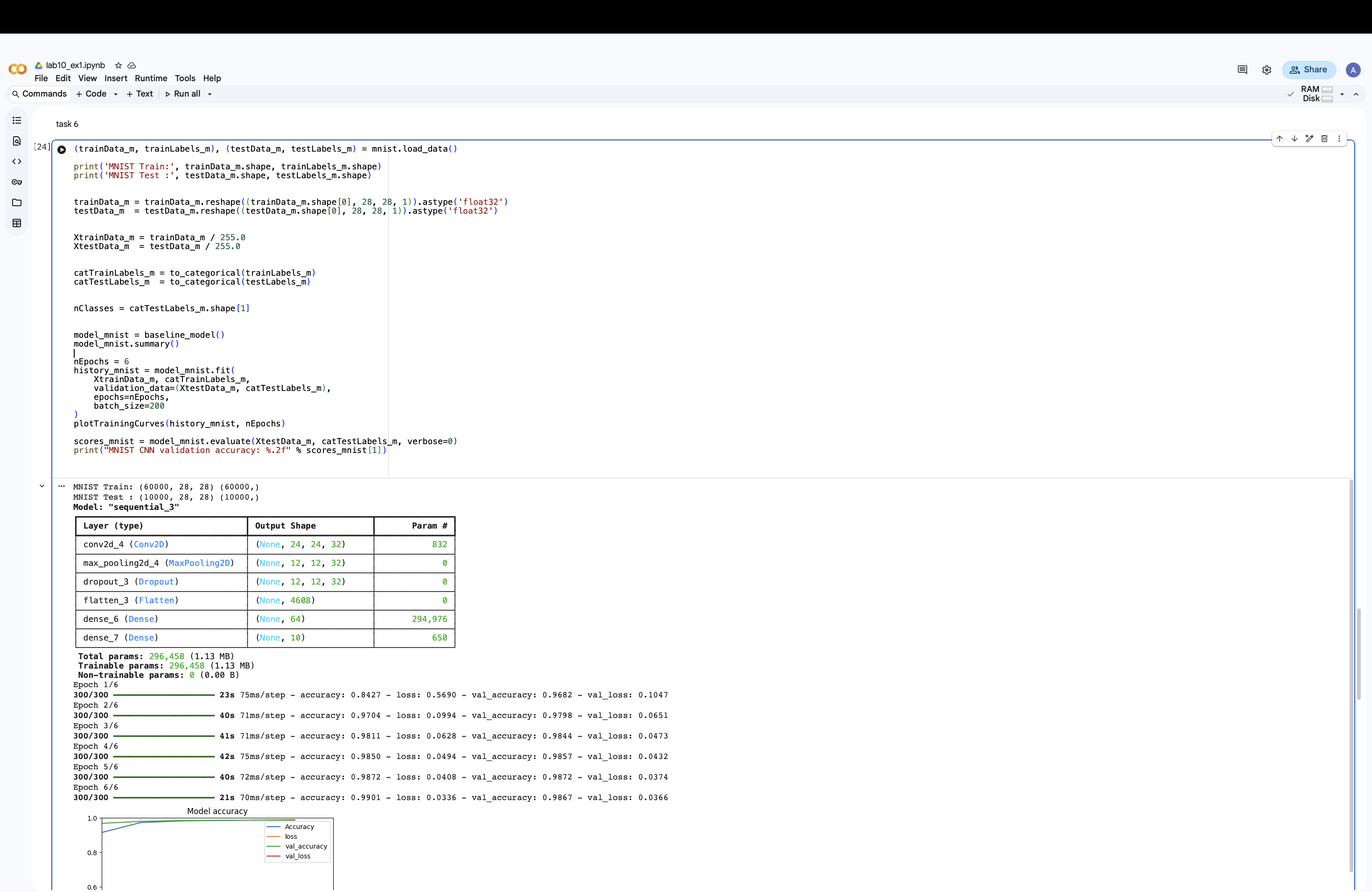
The original model reaches a validation accuracy of about 0.89 with a final validation loss of 0.31, while the enhanced model reaches a similar validation accuracy of about 0.89 but with a slightly lower and more stable validation loss (~0.30). This shows that the second model generalizes at least as well as the first despite having far fewer parameters. The smoother val\_loss curve indicates that the second model learns more efficiently and avoids unnecessary model complexity.

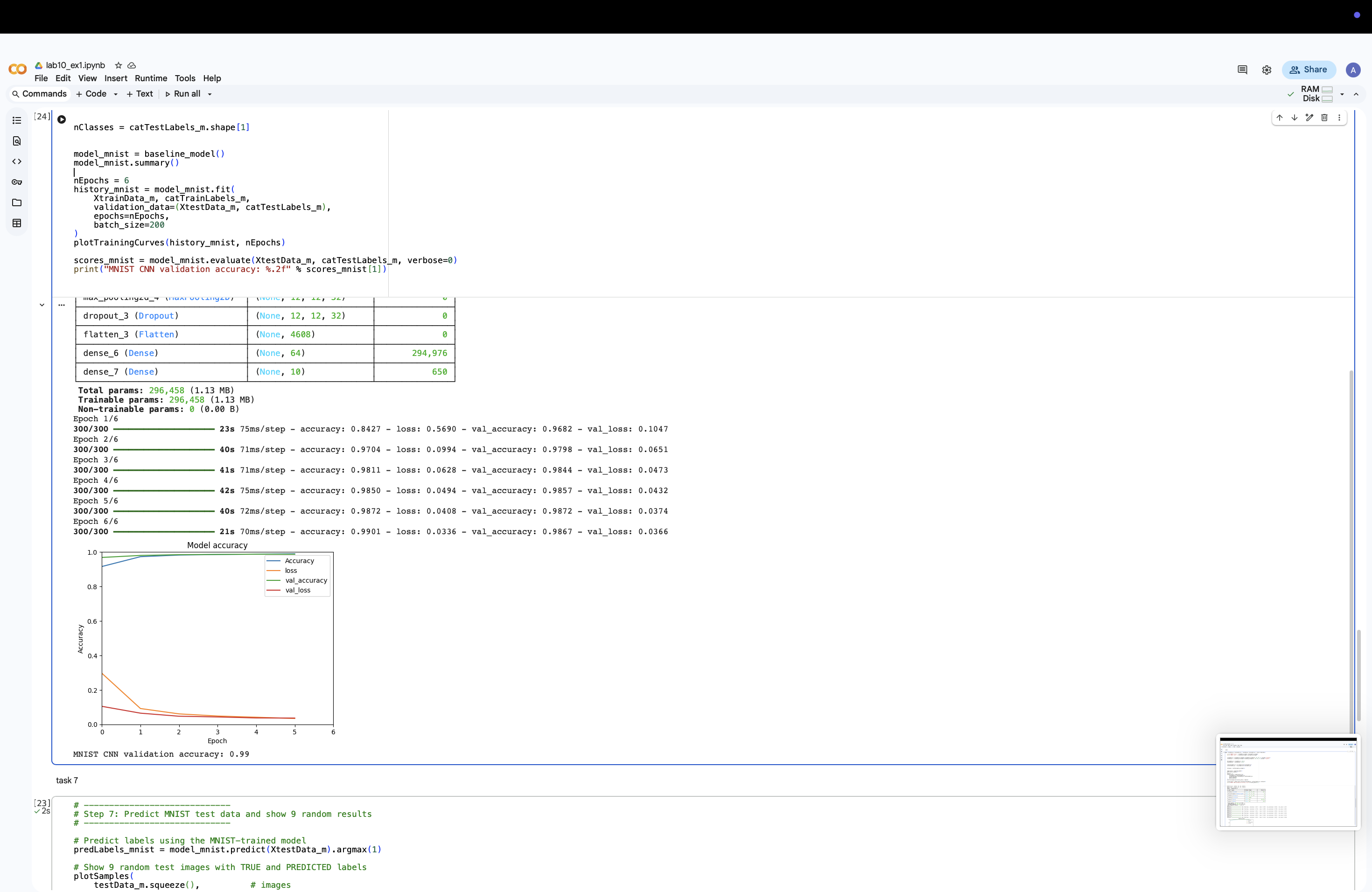




## Step 6 – Training on MNIST Dataset

The same CNN model was trained on the MNIST digits dataset to test generalization. MNIST is visually simpler than Fashion-MNIST, so the model achieved higher validation accuracy.





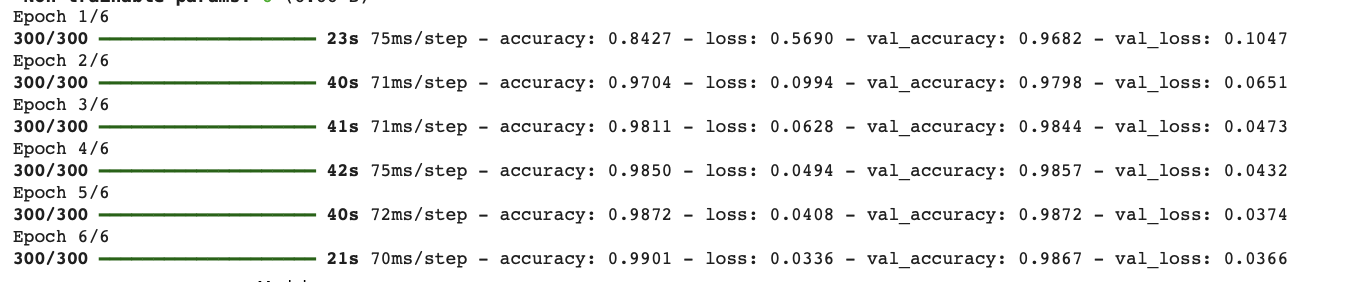
## Step 7 – MNIST Prediction Visualization

Nine random MNIST test images were predicted and displayed with their true labels. Most digits were classified correctly.



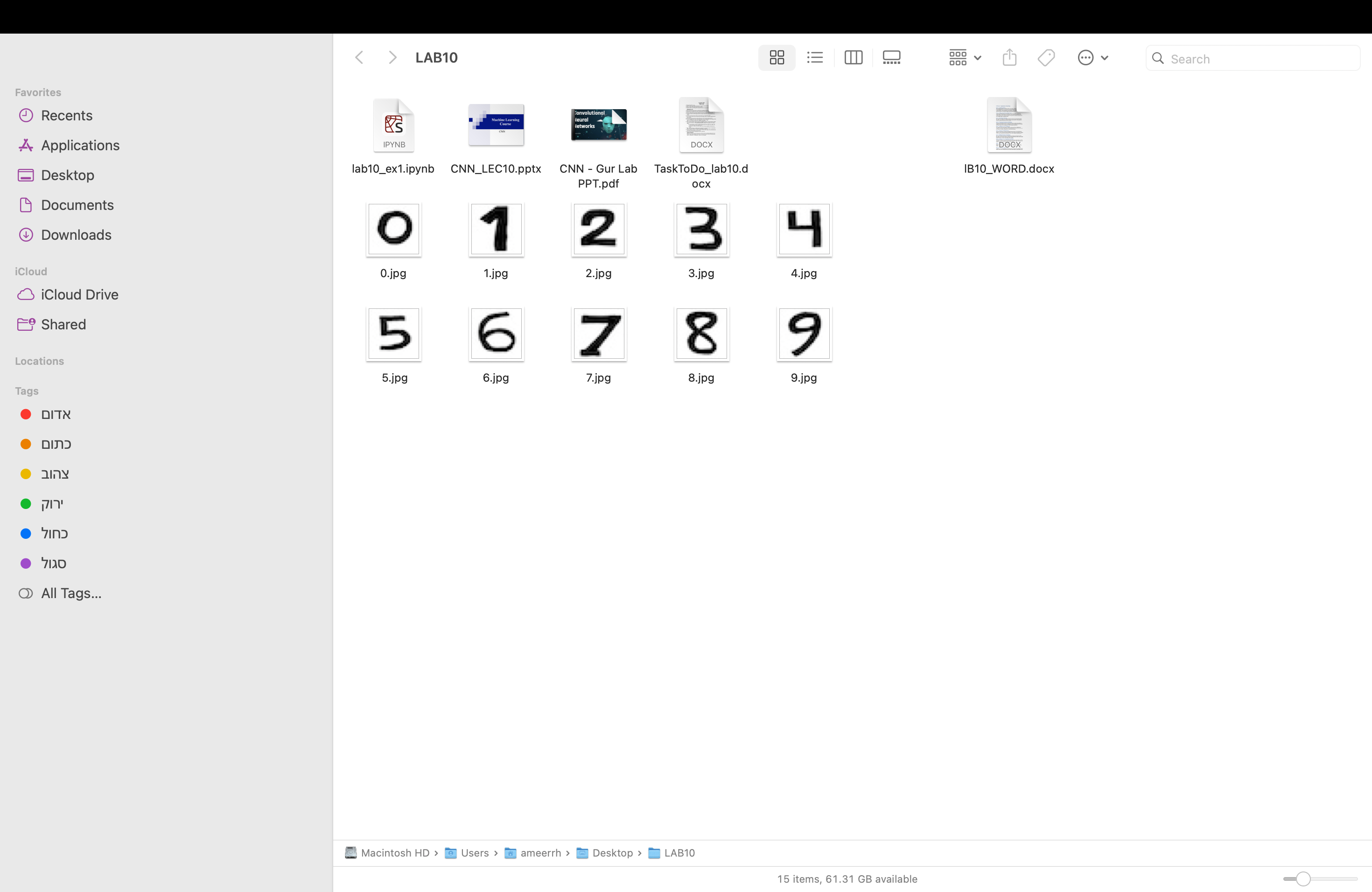
## Step 8 – Validation Curve Explanation for MNIST

The MNIST model achieves a very high validation accuracy of about 0.99, which indicates that the network correctly classifies almost all unseen digit images. At the same time, the validation loss decreases steadily to around 0.03, showing that the model becomes more confident in its predictions. The small gap between training accuracy and validation accuracy suggests that the model generalizes well and does not suffer from overfitting. These curves explain why the visual predictions in Step 7 are mostly correct.



## Step 9 – Creating Handwritten Digits

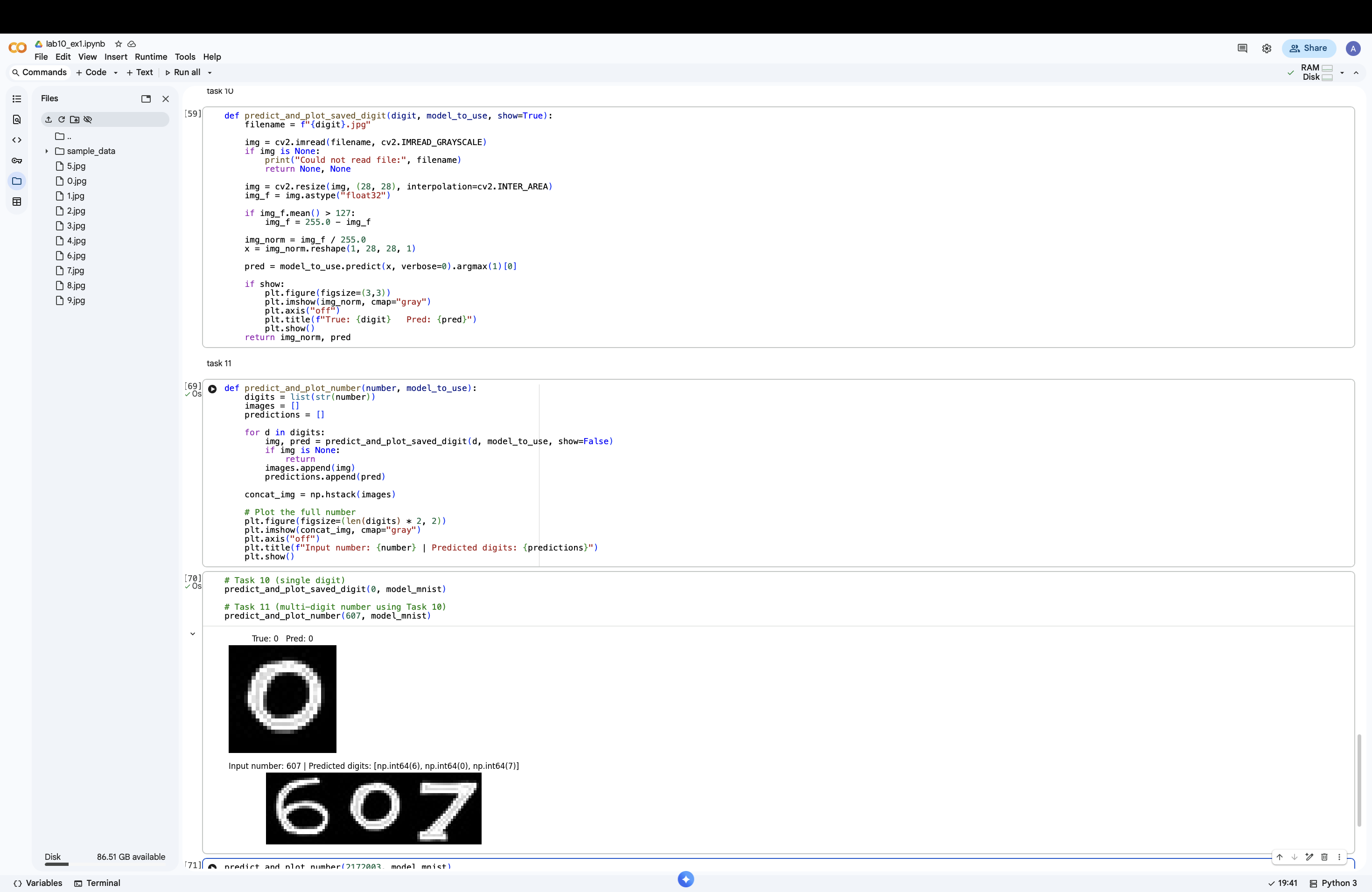
Handwritten digits were created manually using a drawing application and saved as images. Each image was resized to 28×28 pixels to match MNIST format. This step simulates real-world input outside the dataset.



## Step 10 + 11– Predicting Custom Digits + Drawing Multi-Digit Numbers

A function was used to load handwritten digit images and predict them using the trained model. Images were normalized and color-inverted when needed to match MNIST style. This ensured that all predictions were accurate.

A function was used to combine multiple handwritten digit images into one long image. Each digit was predicted separately and then concatenated.



## Step 12 – Displaying Date of Birth

The function from Step 11 was used to display the date of birth using handwritten digits. Day, month, and year were represented in one single line.

