Machine Learning

Corrado Valeri, Michele Baldo

next slide →

02

Our Goals

Our goal is to train a CNN model to classify the fonts based on 1250 images

→

03

Methods

Data Augmentation

- Random Affine = Applied random translation of up to 10%
- Random Flip = Random horizontal and vertical flips
- ColorJitter = Random changes in brightness and contrast

In practice: you take an image and transform it in other ways → so the model learns to recognize characters/fonts even if the image is rotated, enlarged, slightly deformed, etc.

Data Preprocessing

- Grayscale Conversion = The images are converted to grayscale using PIL to reduce complexity
- Binarization = Each image is binarized by applying a threshold (in our case 128) to create a high contrast image
- Denoising = A median filter is applied to remove noise from the binary image

next slide →

04

CNN Architecture (ResNet 50)

TRANSFER LEARNING

*We used **ResNet50**, a model pre-trained on **ImageNet**. The model is modified by replacing its final classification layer to match the number of classes in our dataset (the number of unique fonts).*

TRAINING

We fine-tuned the last fully connected layer while freezing the weights of the pre-trained layers to retain learned features.

LOSS FUNCTION

Cross Entropy Loss = was used for multi-class classification

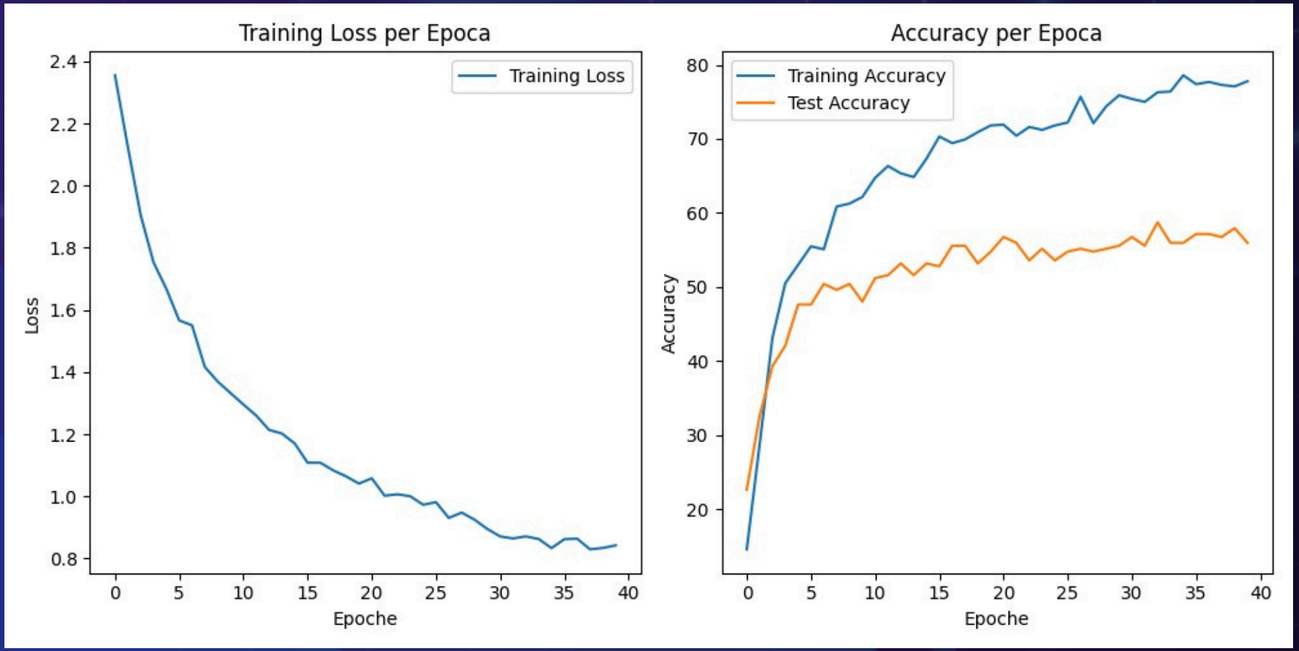
OPTIMIZER

SDG= with a learning rate of 0.001 and momentum 0.9

05 Metrics and Results - 1 attempt

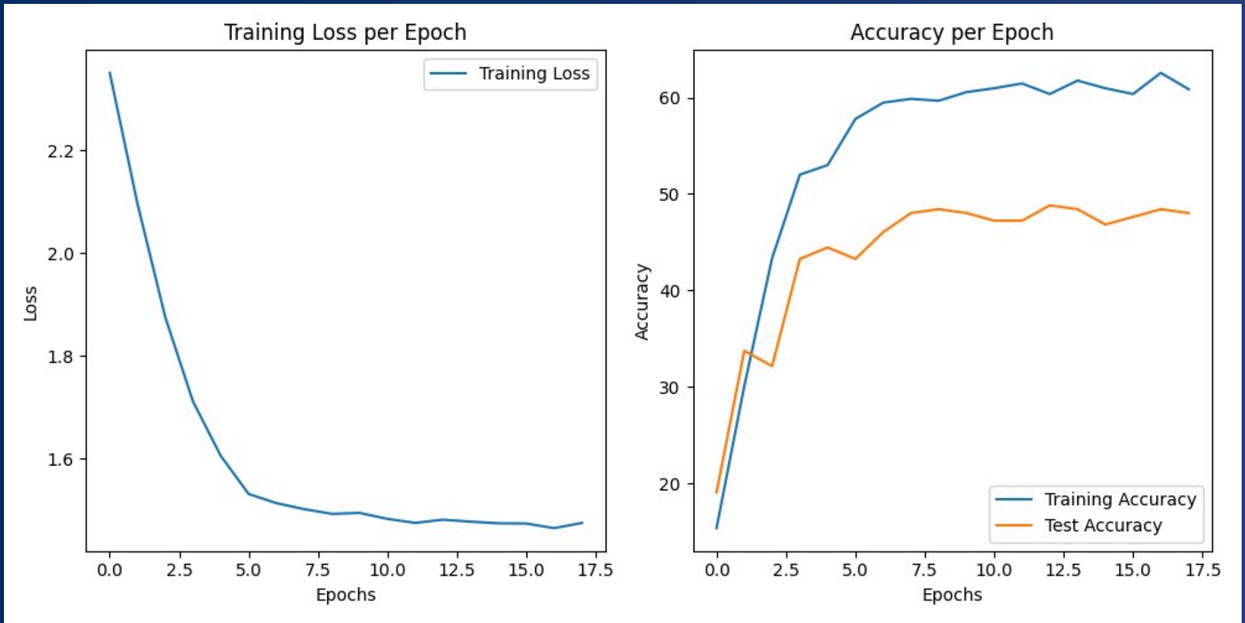
- The training loss decreased steadily with each epoch, indicating that the model was learning the features associated with the fonts.

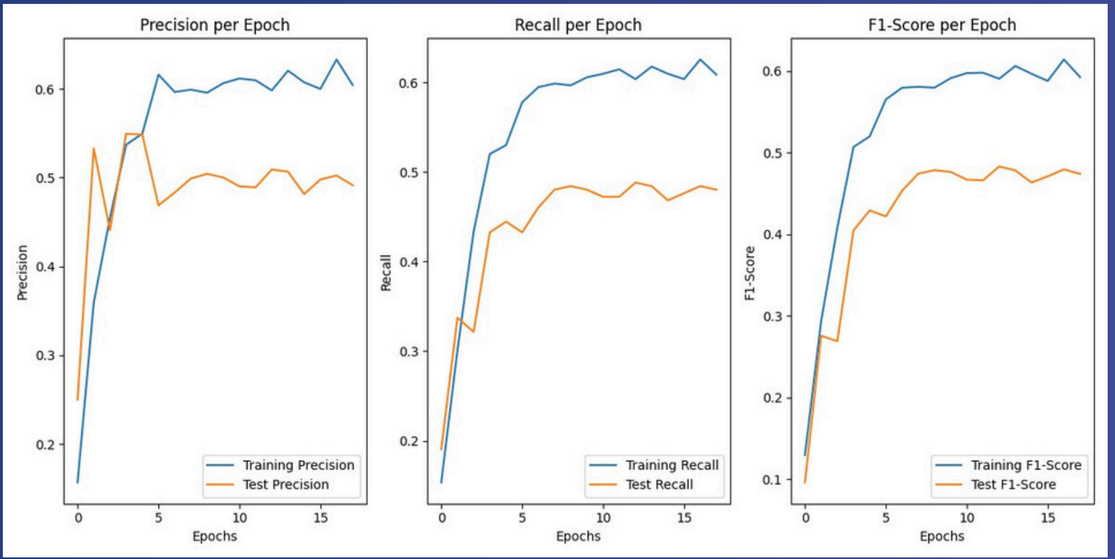
- The accuracy for the training set increased, showing that the model was able to correctly classify the fonts in the training data.



Epoch	Training Los	ining Accur	est Accuracy
1	2,354755	14,54183	22,61905
2	2,124416	28,68526	32,53968
3	1,904904	43,12749	39,28571
4	1,753001	50,49801	42,06349
5	1,6678	52,98805	47,61905
6	1,565995	55,47809	47,61905
7	1,550268	55,07968	50,39683
8	1,416092	60,85657	49,60317
9	1,369393	61,25498	50,39683
10	1,332276	62,15139	48,01587
11	1,295598	64,74104	51,19048
12	1,259922	66,33466	51,5873
13	1,213883	65,33865	53,1746
14	1,202156	64,84064	51,5873
15	1,169949	67,33068	53,1746
16	1,108238	70,31873	52,77778
17	1,108122	69,42231	55,55556
18	1,083448	69,92032	55,55556
19	1,064313	70,91633	53,1746
20	1,041016	71,81275	54,7619
21	1,058005	71,91235	56,74603
22	1,002101	70,41833	55,95238
23	1,006575	71,61355	53,57143
24	1,000236	71,21514	55,15873
25	0,973051	71,81275	53,57143
26	0,981078	72,21116	54,7619
27	0,930533	75,69721	55,15873
28	0,947702	72,11155	54,7619
29	0,92509	74,40239	55,15873
30	0,895033	75,89641	55,55556
31	0,870977	75,39841	56,74603
32	0,864452	75	55,55556
33	0,871067	76,29482	58,73016
34	0,862692	76,39442	55,95238
35	0,833524	78,58566	55,95238
36	0,862156	77,39044	57,14286
37	0,863666	77,68924	57,14286
38	0,82949	77,29084	56,74603
39	0,833857	77,09163	57,93651
40	0,842276	77,78884	55,95238

METRICS AND RESULTS - 2 ATTEMPTS

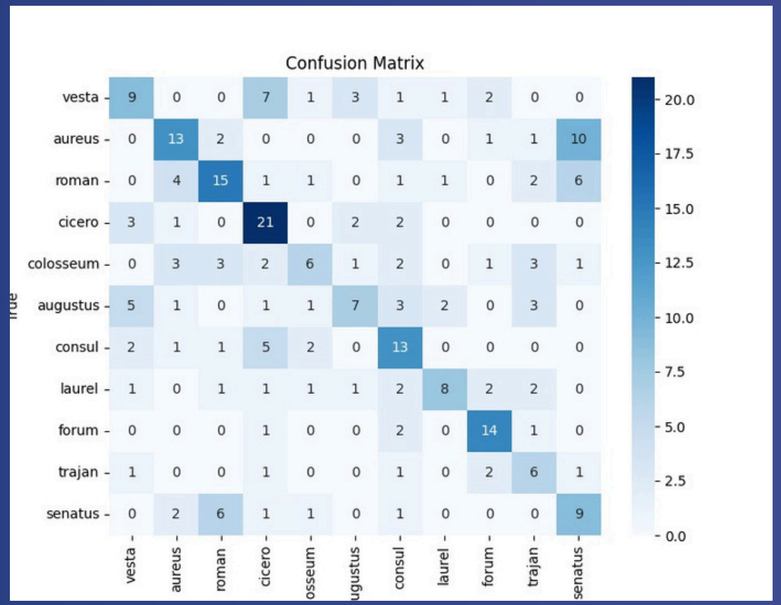


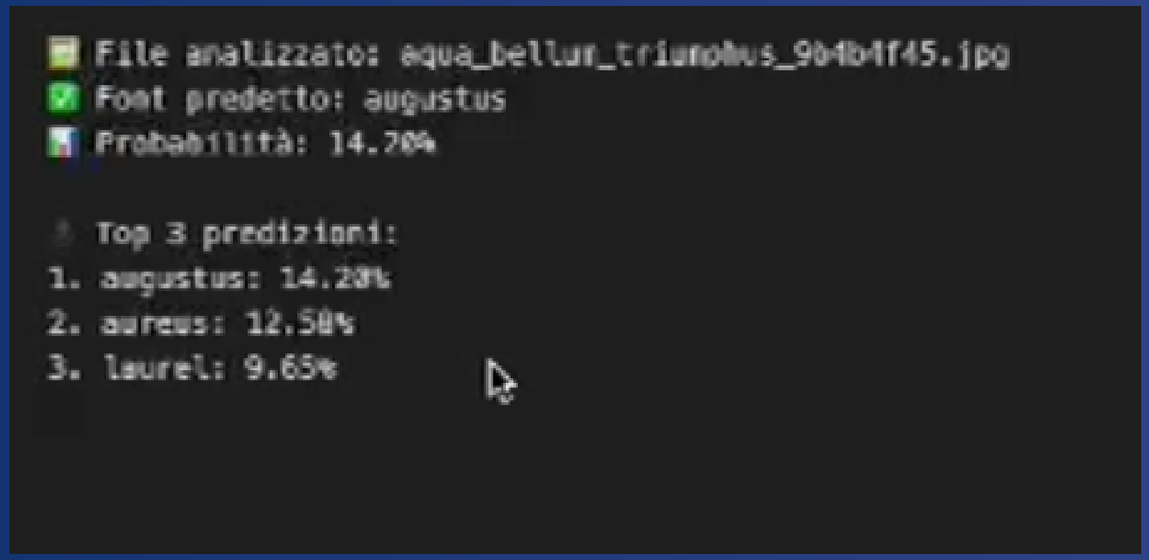


IN THE 2ND ATTEMPT WE USED THE SCHEDULER TO REDUCE THE LEARNING RATE (THE LEARNING SPEED) AFTER SOME TIME, SO THE MODEL DOES NOT "JUMP" TOO MUCH AND STABILIZES BETTER.

IN THIS CASE:
STEP_SIZE=5: EVERY 5 EPOCHS THE LEARNING RATE IS LOWERED
GAMMA=0.1: THE LEARNING RATE IS MULTIPLIED BY 0.1 (I.E. IT IS REDUCED BY 90%).

THE MODEL SHOWS GOOD CLASSIFICATION CAPABILITIES FOR LABELS SUCH AS CICERO, ROMAN, FORUM AND CONSUL, WITH A HIGH NUMBER OF CORRECT PREDICTIONS (DIAGONAL VALUES).
CONFUSIONS PRESENT BUT BALANCED: LESS RECOGNIZED CLASSES SHOW DISTRIBUTED ERRORS (E.G. VESTA, AUGUSTUS), INDICATING THAT THE MODEL DOES NOT LOCK ON SPECIFIC CLASSES.





Font Classifier with Image Selection Interface: This project uses a neural network (ResNet50) to automatically identify the font contained in an image. Through a graphical interface, you can select a .jpg or .png file from your computer and get:

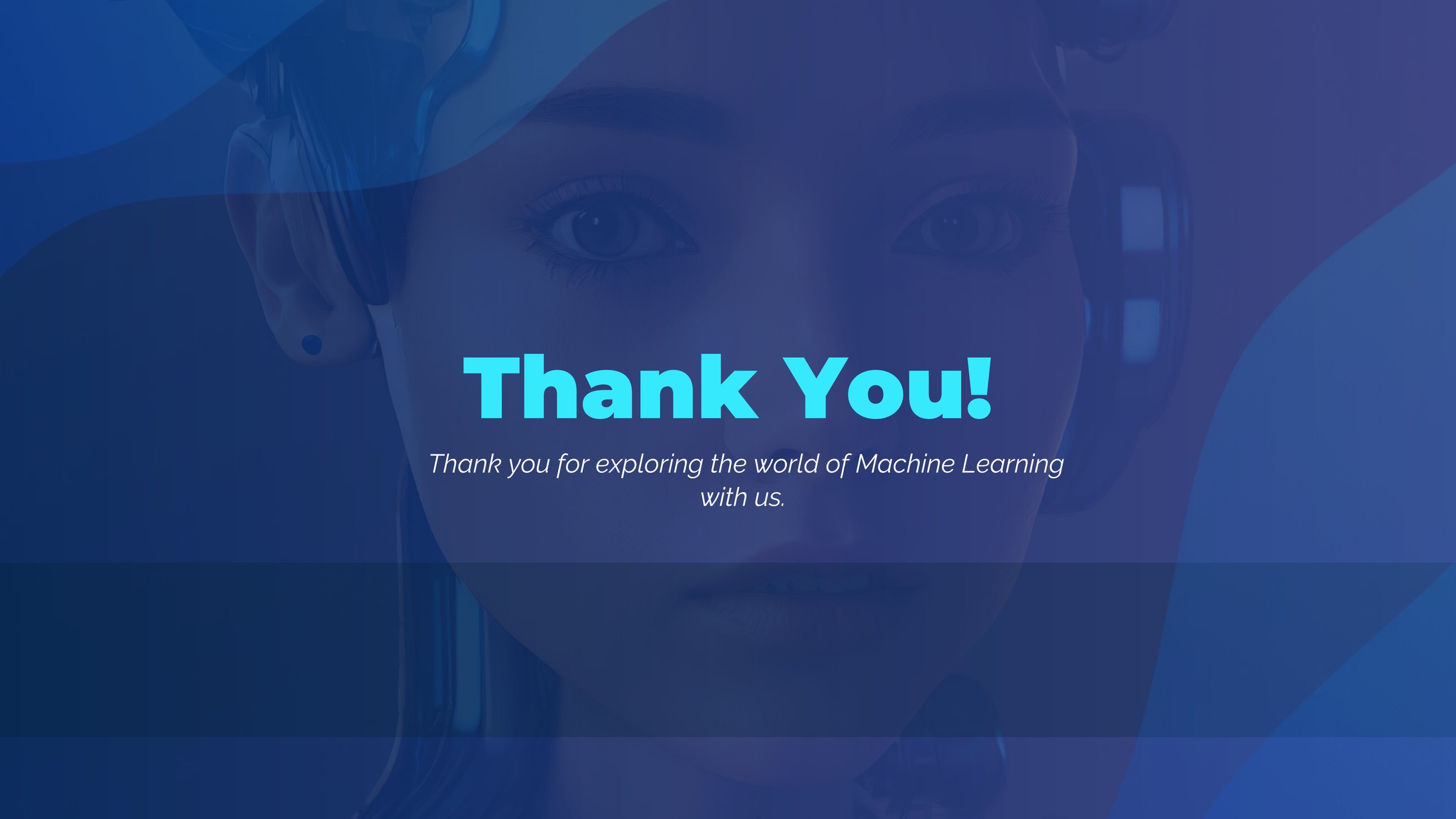
- The name of the detected font
- The probability (confidence) of the prediction
- The Top 3 predictions with their percentages

Conclusions

*This project successfully applied CNNs with Transfer Learning to classify image font types. The model performed well, with a final test accuracy of **56%**. By using data augmentation and fine-tuning a pre-trained ResNet50 model, we were able to achieve significant performance improvements with relatively small data.*

CHALLENGES

- **INCREASING DATASET SIZE**: EXPANDING THE DATASET COULD HELP IMPROVE THE GENERALIZATION ABILITY OF THE MODEL.
- **IMPROVED DATA AUGMENTATION**: EXPERIMENTING WITH ADDITIONAL DATA AUGMENTATION TECHNIQUES, SUCH AS MORE ADVANCED IMAGE TRANSFORMATIONS, COULD FURTHER BOOST MODEL PERFORMANCE.
- **EXPLORING OTHER CNN ARCHITECTURES**: USING DEEPER MODELS OR LIGHTWEIGHT MODELS LIKE **MOBILENET** COULD BE CONSIDERED FOR BETTER EFFICIENCY.



Thank You!

Thank you for exploring the world of Machine Learning with us.