Ml project

*I declare that this material, which I now submit for assessment, is entirely my own work and has not been taken from the work of others, save and to the extent that such work has been cited and acknowledged within the text of my work. I understand that plagiarism, collusion, and copying are grave and serious offences in the university and accept the penalties that would be imposed should I engage in plagiarism, collusion or copying. This assignment, or any part of it, has not been previously submitted by me or any other person for assessment on this or any other course of study*

**Quindi nel report devi includere**:

* **Quali iperparametri** hai ottimizzato
* **Quali range** di valori hai testato
* **Quale metodo** hai usato (grid search, CV, ecc.)
* Eventuali **grafici** delle performance in funzione degli iperparametri
* **Qual è il valore ottimale** scelto e perché
* Essere ben strutturato (introduzione → metodo → risultati → discussione → conclusioni)
* ghp\_kbZBIPGmMUpIUlv1KCpvHUutRNsDlr1vQj3H

git add Samuele.Magatti.ML.Project.py

git commit –m “end of eda, start preprocessing”

git push

#ML project

import glob

images = glob.glob("data/\*.jpg") + glob.glob("data/\*.png")

if len(images) == 0:

raise FileNotFoundError("No images found in data/ (expected .jpg or .png files)")

#Data exploration

# Preprocessing

## image resizing

## normalization

## data augmentation

## split data into train and test set

required\_files = ["data/train.csv", "data/test.csv"]

for f in required\_files:

if not os.path.exists(f):

raise FileNotFoundError(f"Missing {f}. Put the data in a folder called 'data/' as written in the README file")

#build of first CNN

Mean: R=0.3229, G=0.5491, B=0.2601

Std: R=0.0514, G=0.0439, B=0.0552

Train samples: 1531

Validation samples: 328

Test samples: 329

Found 4376 PNG files.

Total corrupted images: 0

Total duplicates found: 0

{'paper': 712, 'rock': 726, 'scissors': 750}

paper: 712 file

rock: 726 file

scissors: 750 file

filename width height class

0 W79peyAyfQqNP1vF.png 300 200 paper

1 7ZGsbIMypDIEZxR0.png 300 200 paper

2 DsaaqkHZUN3pmcwH.png 300 200 paper

3 P1STllnKDD05zo29.png 300 200 paper

4 RlpqfQnTgM6AxUNs.png 300 200 paper

filename width height class aspect\_ratio

0 W79peyAyfQqNP1vF.png 300 200 paper 1.5

1 7ZGsbIMypDIEZxR0.png 300 200 paper 1.5

2 DsaaqkHZUN3pmcwH.png 300 200 paper 1.5

3 P1STllnKDD05zo29.png 300 200 paper 1.5

4 RlpqfQnTgM6AxUNs.png 300 200 paper 1.5

total images controlled: 2188

non RGB images: 0

Mean: R=0.3229, G=0.5491, B=0.2601

Std: R=0.0514, G=0.0439, B=0.0552

Train Mean: R=0.3220, G=0.5481, B=0.2593

Train Std: R=0.2556, G=0.1014, B=0.1329

Train samples: 1531

Validation samples: 328

Test samples: 329

SimpleCNN(

(conv1): Conv2d(3, 32, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1))

(conv2): Conv2d(32, 64, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1))

(conv3): Conv2d(64, 128, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1))

(pool): MaxPool2d(kernel\_size=2, stride=2, padding=0, dilation=1, ceil\_mode=False)

(fc1): Linear(in\_features=118400, out\_features=256, bias=True)

(fc2): Linear(in\_features=256, out\_features=3, bias=True)

(dropout): Dropout(p=0.5, inplace=False)

)

Epoch 1/10 | Train Loss: 0.8217 | Train Acc: 0.6773 | Val loss: 0.3340 | Val Acc: 0.9024

Epoch 2/10 | Train Loss: 0.2114 | Train Acc: 0.9301 | Val loss: 0.1857 | Val Acc: 0.9421

Epoch 3/10 | Train Loss: 0.1070 | Train Acc: 0.9660 | Val loss: 0.1822 | Val Acc: 0.9512

Epoch 4/10 | Train Loss: 0.0884 | Train Acc: 0.9713 | Val loss: 0.2007 | Val Acc: 0.9421

Epoch 5/10 | Train Loss: 0.0409 | Train Acc: 0.9863 | Val loss: 0.1406 | Val Acc: 0.9665

Epoch 6/10 | Train Loss: 0.0256 | Train Acc: 0.9895 | Val loss: 0.1508 | Val Acc: 0.9604

Epoch 7/10 | Train Loss: 0.0279 | Train Acc: 0.9928 | Val loss: 0.1597 | Val Acc: 0.9695

Epoch 8/10 | Train Loss: 0.0313 | Train Acc: 0.9889 | Val loss: 0.1864 | Val Acc: 0.9665

Epoch 9/10 | Train Loss: 0.0314 | Train Acc: 0.9902 | Val loss: 0.2013 | Val Acc: 0.9604

Epoch 10/10 | Train Loss: 0.0117 | Train Acc: 0.9961 | Val loss: 0.1366 | Val Acc: 0.9817

Test Accuracy: 0.9726

“Per valutare l’impatto della complessità architetturale, è stato confrontato il modello ottimizzato con una CNN semplificata (2 blocchi convoluzionali). Il confronto mostra che la rete intermedia offre un miglior trade-off tra accuratezza e costo computazionale.”