MACHINE LEARNING PROJECT

Samuele Magatti

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.

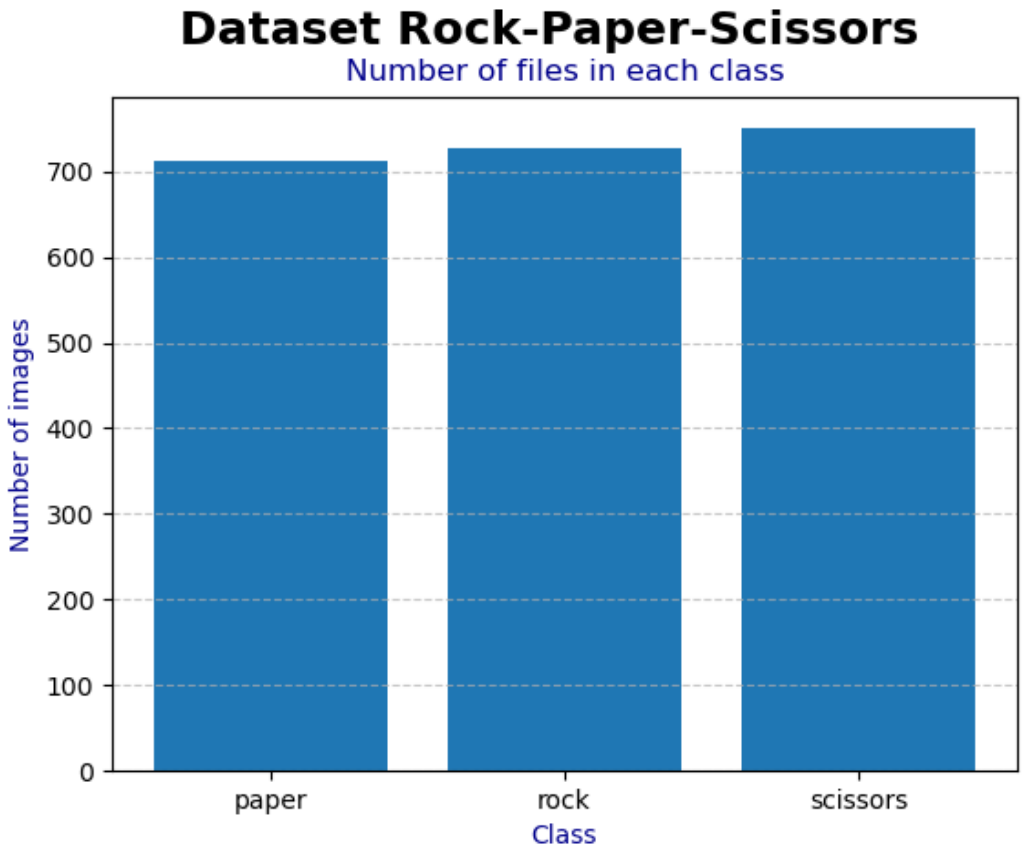
Abstract

The aim of this project is to implement a classification task with three different covolutional neural networks algorithms built with incremental complexity. Using the Rock-Paper-Scissor dataset, composed of images representing human hands on a green screen with three different hand postures, the three neural networks are supposed to be able to correctly assign a label, rock paper or scissor, to each of the images. An additional objective is to observe wheather the data augmentation tecniques implemented in the preprocessing are able to product predictors able to generalize their results when using images from other sourches.
The images were divided into three parts, training validation and test, for the first two algorithms, for the training part it has been used data augmentation tecniques while the normalization using only the data from the training part was done for all the three parts. For the last neural network the dataset was split into train and test part and it was implemented a grid search cross validation to select the best hyperparameters for the neural network.
To compare the three neural networks were used the following metrics: accuracy, precision, recall, f1 score confusion matrix, it was also used a label shuffle test to prevent eventual data leakage, furthermore it has been used an external homemade dataset with a different background to check external validity.
After the data exploration and preprocessing tecniques has been implemented the third neural network has reached a final accuracy of 99% on the test dataset and 33% on the external dataset, this pattern in accuracy is present in all the three neural networks underlying the fact that the algorithms have a problem in generalizing probably linked with different backgrounds and brightness.

Explorative data analisys results

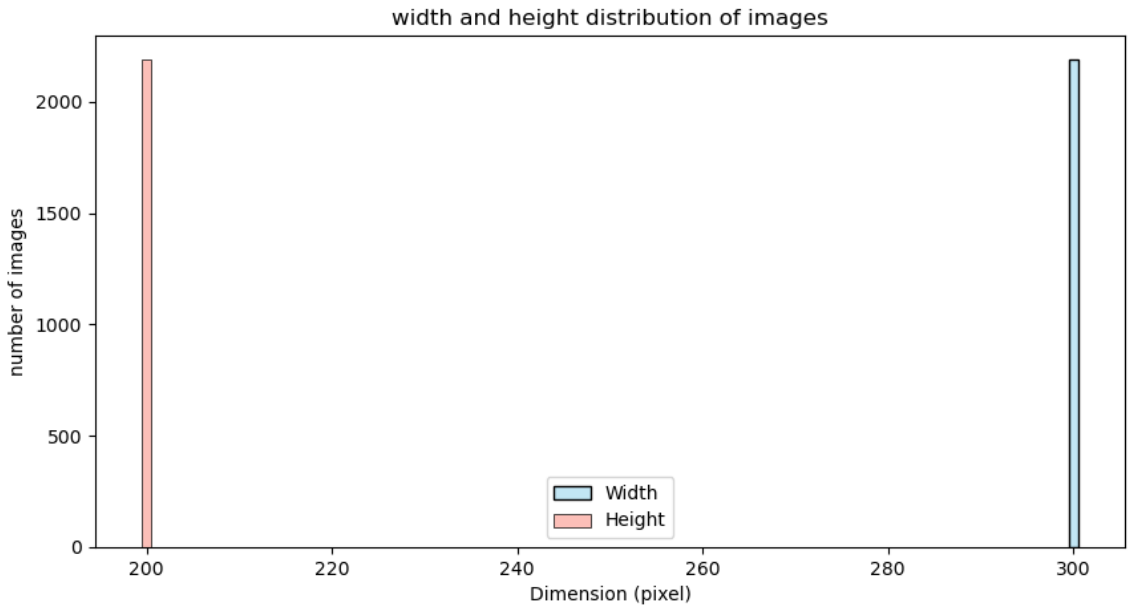
Having verified the absence of dublicate or corrupted files the code shows that the dataset contains 2188 images, 712 are classified as paper, 726 are classified as rock and the remaining 750 are in the scissor label, thus the three classes seems to be balanced with around 700 examples each.

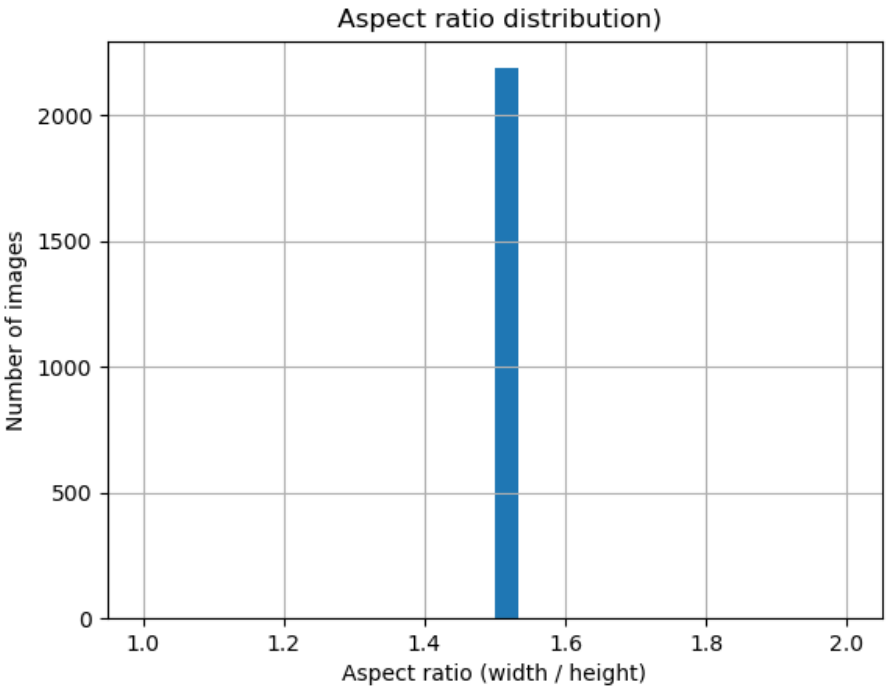
The slightly difference (from 710 paper, 726 rock and 752 scissor to 712, 726 and 750 respectively) between what is written in the readme of the dataset and the resulst is probabli due to the fact that the latter was probably modified to made the classes distribution more equal.



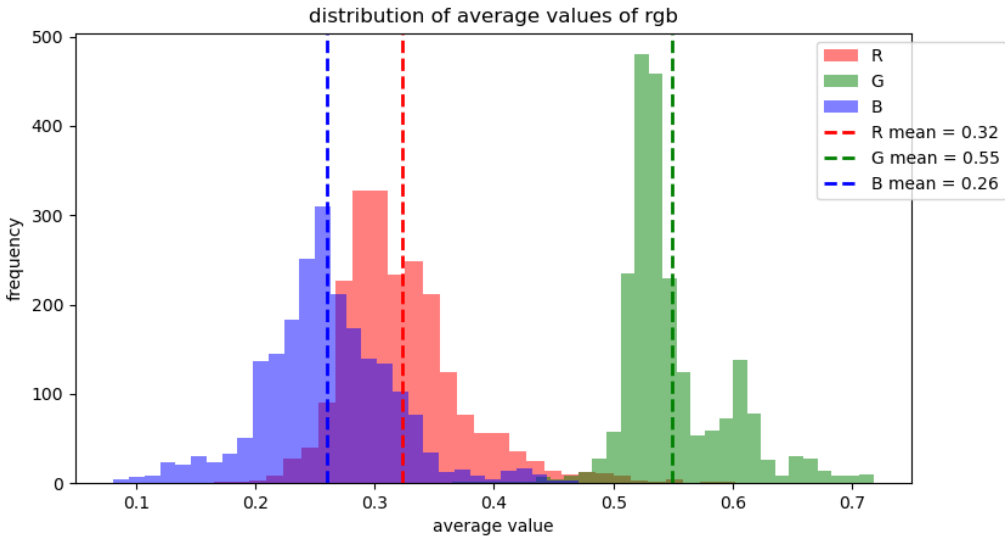
.

All the images are in the same .png formet and have uniform dimensions (300 x 200) and the same aspect ratio of 1.5.





The color distribution analysis, that can be seen in the histogram below, shows that the green channel has consistently higher average values, which is expected given the green background used for all photos. Consequently, data augmentation operations that alter color saturation were avoided to preserve consistency across samples.



Data Preprocessing

Two different pipelines has been used for preprocessing the data for the neural networks. For the first two algorithms 70% of the images, corresponding to 1531 images, were put in the train dataset, while the train sample and the validation sample were 15% each with 328 and 329 images respectively. All the images were resized to 150x225, thus mantaining the aspect ratio of 1.5, and were normalized using only the data of the train sample. Only for the data of the train sample hava been implemented data augmentation tecniques such as horizontal flip with probability of 50%, random rotation of 15 degrees and changes of contrast and brightness.

On the other hand for the third neural network, since it was necessary to implemente a grid search cross validation, different tecniques have been used. The dataset was divided into training set, 80 % and 1750 images, and test set, the remaining 20% and 438 images. The train set has been sequently divided into 3, for running time reasons, folders and the mean and standard deviation have been calculated for the first two folds used for the train and used for the normalization of the validation and train part to avoid any data leakage. The previously defined data augmentation tecniques were implemented also for the train part of the third cnn, to have a more robust predictor.
When using the external validity homemade dataset the same normalization values computed for the train set have been used.

First C Neural Network

Architecture

The first implemented neural network is a Tiny-CNN, thought to evaluate the outputs of a simple covolutional model on the uppersaid dataset.
The architecture is made by:

- 2 blochs Conv2D + ReLU + MaxPooling, with 16 and 32 filtersris respectively
- dropout (0.3) to avoid overfitting
- 1 intermediate layer Fully Connected with 64 neuroni
- 1 final layer FC with 3 neurons one for each class

The network uses a relu activation function and cross-entropy loss as loss function.

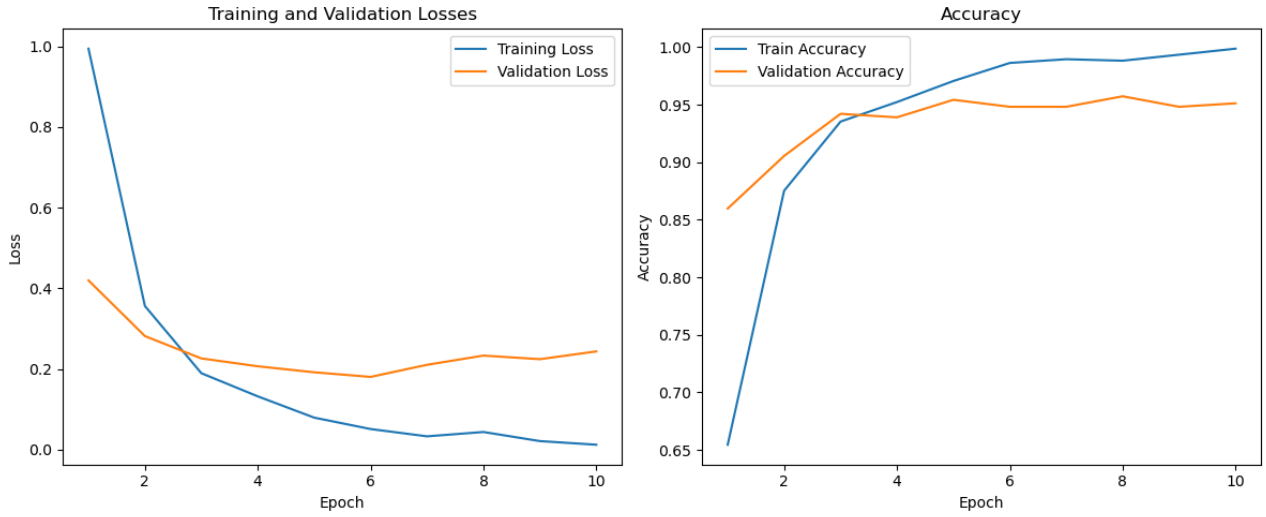
The choosen optimizar is Adam (learning rate 0,001).

Layer	Output Shape	Paramethers
Conv2D (3→16) + ReLU	200×300	448
MaxPool 2×2	100×150	–
Conv2D (16→32) + ReLU	100×150	4,640
MaxPool 2×2	50×75	–
Flatten	32×50×75	–
FC (120000→64)	64	7.680.064
Dropout 0.3	64	–
FC (64→3)	3	195

Training and validation

The model has been trained for ten epochs using augmentation and splitting the dataset in train validation and test as previously said. During the learning the loss and accuracy for both the training and validation sets have been monitored, as can be seen in the figures. In the first epochs both train loss and accuracy are worse than validation loss and accuracy, this is mainly due to the data augmentation tecniques, in fact for the model is more difficult to assign labels in the data without augmentation. After the first epochs the validation values seem to stabilize while the train loss and accuracy continue increasing. This indicates the onset of overfitting, as the model continues to

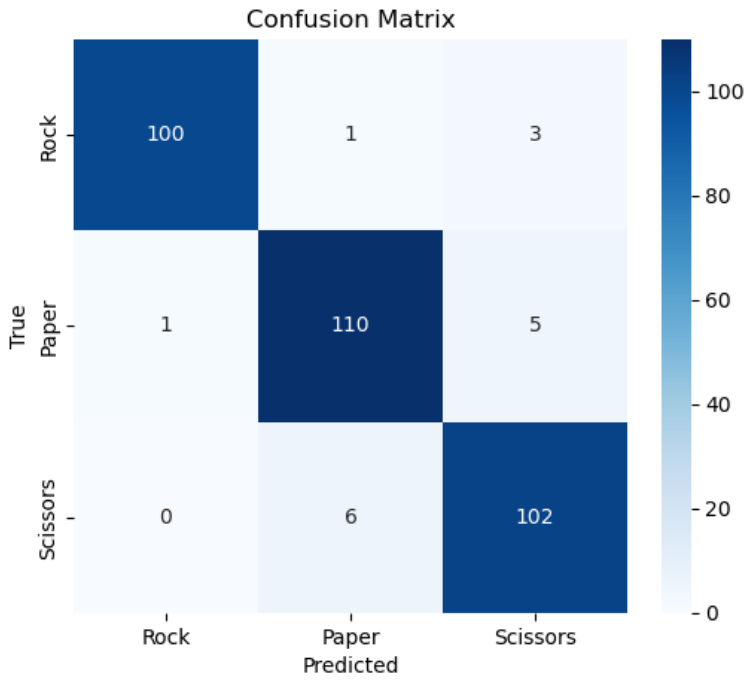
improve on the augmented training data without achieving further generalization on the validation set.
Although the global trend of the curves is stable, small variations may appear between repeated runs. These differences are mainly due to nondeterminism in data loading and the absence of a fully fixed random seed on macOS, which can result in different minibatch orders across runs.



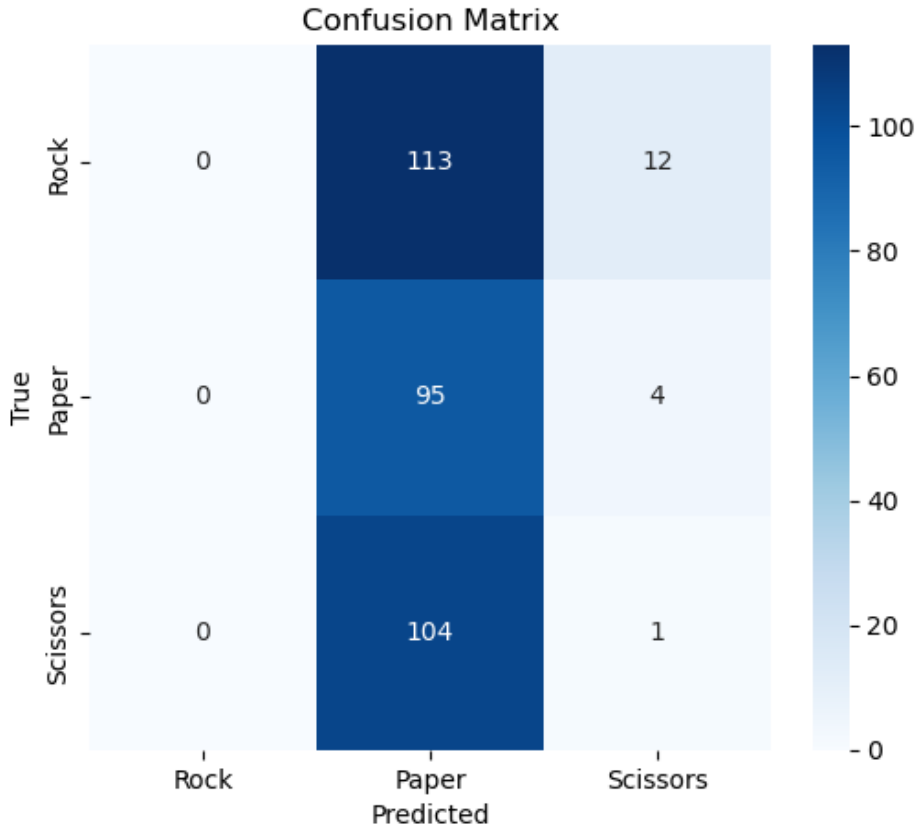
Results and conclusion

On the test set the neural network reaches an accuracy of 96,35%, the confusion matrix underlines the fact that the model is able to predict well the labels of the original dataset since the diagonal has higer values. All the metrics seem to be high, underliìying the fact that the model is able to predict well the labels in the dataset.

CLASS	PRECISION	RECALL	F1	SUPPORT
0	0.9915	0.9360	0.9630	125
1	0.9697	0.9697	0.9697	99
2	0.9286	0.9905	0.9585	105
Accuracy			0.9635	329
Macro avg	0.9633	0.9654	0.9637	329
Weighted avg	0.9649	0.9635	0.9636	329



To verify the absence of data leakage a lable shuffle test was done, training the same algorithm with casually assigned labels, the result for the two epoch is 32.40% and 33.25%, this with the confusion matrix demonstartes that in this case the model almost always predicts the same class (paper) thus demonstating that there is no data leakage.



Second Covolutional Neural Network

Architecture

To improve the results from the first algorithm a deeper neural network has been implemented named simpleCNN. The architecture is :

- 3 Convolutional blocks + ReLU + MaxPooling with 32, 64 e 128 filters respectively
- MaxPooling 2×2 to reduce the spatial demensions
- An intermediate fully connnected layer made by 256 neurons
- Dropout 0.5 for regularization
- Final FC is made of 3 neurons one for each class.

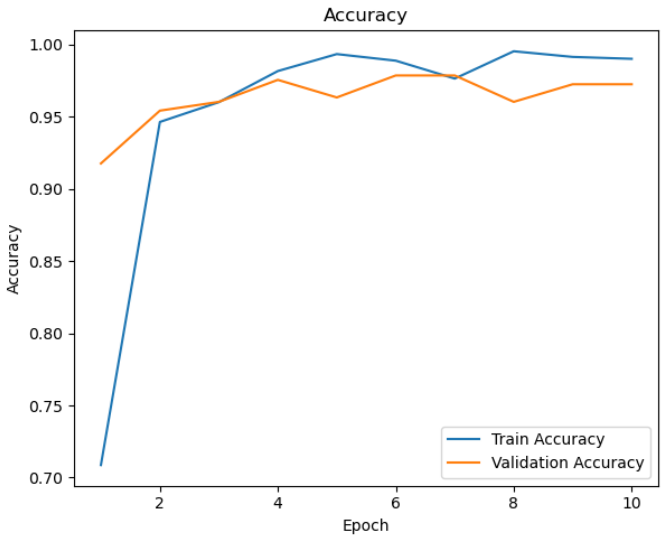
The network uses the cross-entropy loss and the Adam optimizer.

Layer	Output shape	paramethers
Input (rgb)	3x150x225	
Conv2D (3->32) + ReLU	32x150x225	896
MaxPool 2x2	32x75x112	
Conv2D (32>64) + ReLU	64x75x112	18,496
MaxPool 2x2	64x37x56	
Conv2D (64>128) + ReLU	128x37x56	73,856
MaxPool 2x2	128x18x28	
Flatten	64512	
FC (64512>256) + ReLU	256	16,515,328
Dropout (p=0.5)	256	
FC (256>3)	3	771

The majority of the parameter is situated in the first fully connected layer.

Training and validation

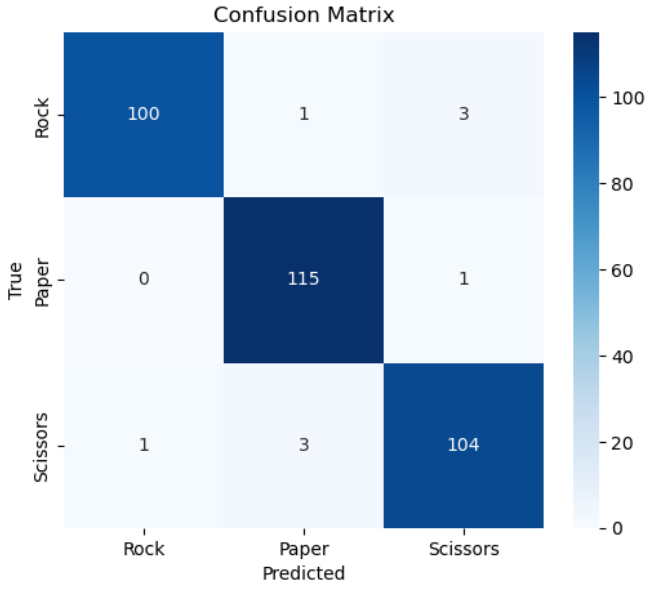
The model has been trained for ten epochs with batch size 32, using augmentation and splitting the dataset in train validation and test as previously said. During the learning the loss and accuracy for both the training and validation sets have been monitored, as can be seen in the figures. As happened for the previous model the train loss and accuracy are worst in the first epochs while in the nd the stabilization of both validation loss and accuracy indicates the starting of overfitting.



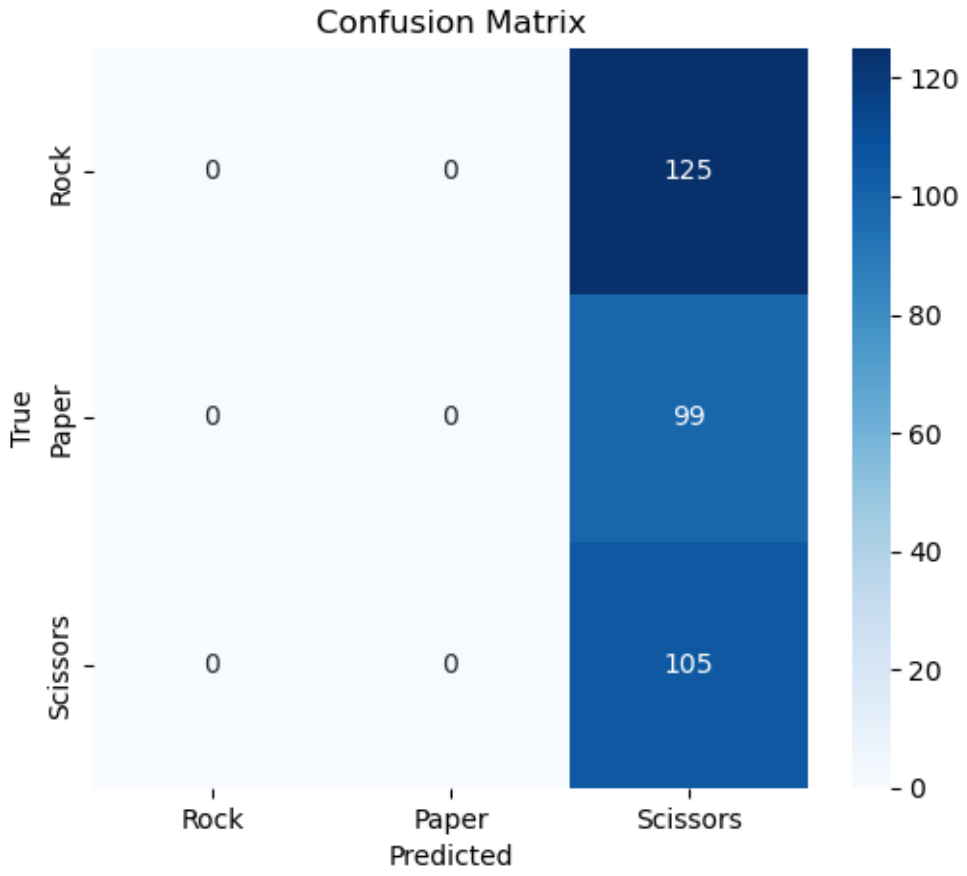
Results and conclusion

On the test set the neural network reaches an accuracy of 96,66%, so it is slightly better than the previous one at a significant computing time differense. The confusion matrix underlines the fact that the model is able to predict well the labels of the original dataset since the diagonal has higer values. All the metrics seem to be high, underliìying the fact that the model is able to predict well the labels in the dataset.

CLASS	PRECISION	RECALL	F1	SUPPORT
0	0.9832	0.9360	0.9590	125
1	0.9510	0.9798	0.9652	99
2	0.9630	0.9905	0.97665	105
Accuracy			0.9666	329
Macro avg	0.9657	0.9688	0.9669	329
Weighted avg	0.9670	0.9666	0.9665	329



To verify the absence of data leakage a lable shuffle test was done, training the same algorithm with casually assigned labels, the result for the two epoch is 34.62% and 35.08%, this with the confusion matrix demonstartes that in this case the model always predicts the same class (scissor) thus demonstating that there is no data leakage.



Third Covolutional Neural Network

Architrcture

In the third experiment, it was designed a deeper and more robust Convolutional Neural Network, hereafter referred to as AdvancedCNN, with the goal of improving generalization and reducing overfitting. Unlike the previous models, this architecture includes a 3-fold Cross-Validation strategy, a Grid Search hyperparameter optimization, and fold-specific normalization statistics. This version represents the most systematic and rigorous training pipeline in the project.

The proposed AdvancedCNN is composed of four convolutional blocks, each containing:

- Two convolutional layers with Batch Normalization and ReLU activation
- A MaxPool layer (only in the first block of each pair)

The network ends with a Global Average Pooling, a Dropout layer, and a fully connected classifier.

Training and grid search CV

Batch Normalization helps stabilize training, while Dropout acts as regularization. The Adaptive Average Pooling drastically reduces the number of parameters compared to Fully Connected layers over flattened feature maps.

A Grid Search was applied over the following hyperparameters:

Hyperparameter	Tested Values
Learning Rate	[1e-3, 1e-4]
Batch Size	[32, 64]
Dropout	[0.3, 0.5]

For each hyperparameter combination:

- The model was trained on all folds
- The weighted F1-score was used as the model selection metric
- The mean F1 across folds determined the best configuration

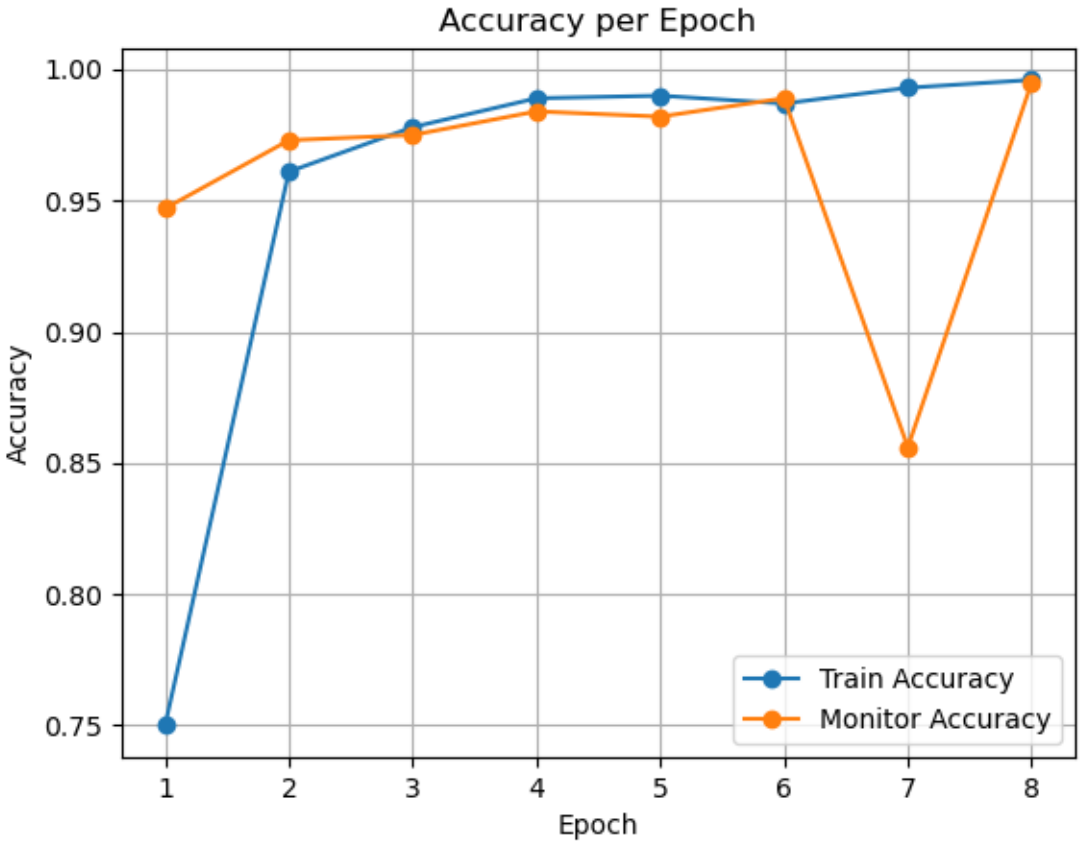
This approach ensures that the chosen parameters generalize well, not just to a single split.

- After the test on all teh eight combination of possible hyperparameters it was found that the best parameters according to the cv mean f1 are 0.0001 for the learning rate, 32 for the batch size and 0.5 for the dropout with a f1 value of 0.969.
- Once the best hyperparameter set was found, the model was retrained on the entire Train/Validation set and then evaluated on the hold-out Test set then the following test and metrics were performed by the code:

- Final Accuracy, Precision, Recall, and F1-Score
- Confusion Matrix
- Visualization of misclassified images
- Training curves (loss and accuracy per epoch)

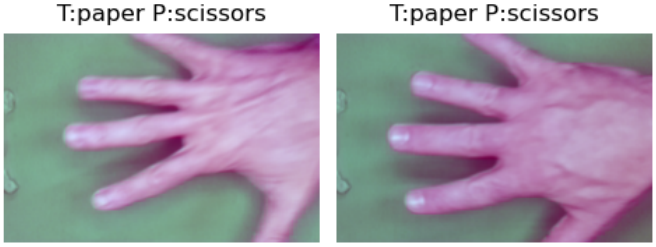
An additional external evaluation was also performed on a separate dataset to measure real-world generalization.

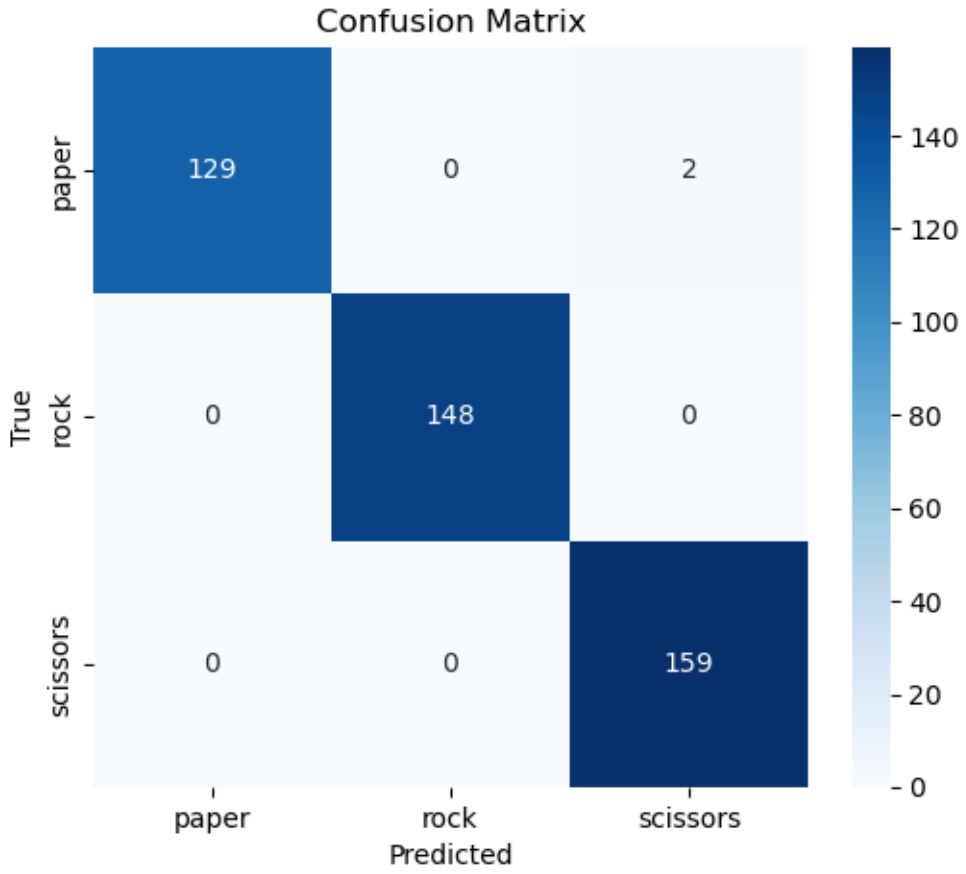
Since it was applied the data augmentation only for the training part also for this model there is a lower training accuracy than monitor, this changes rapidly, in addition to that it is important to note a drop in the seventh epoch, that does not however prevent a stabilization of the overall result since also for the following epochs, that are not represented, the result keeps being stable around 0.98.



Results and conclusion

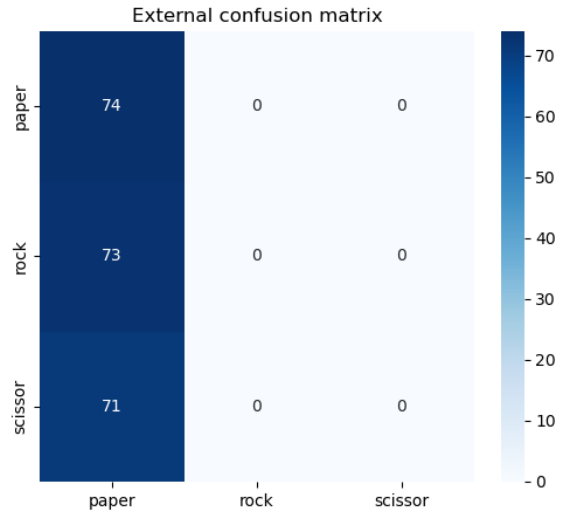
On the test set the neural network reaches an accuracy of 99,54%, mafing a mistake just two times in all the test set. The precisioni s 0.9955, the recall is 0.9954 anf f1 is 0.9954, all these metrics indicates the fact that the model is really able to predict the correct label of the images contained in this specific dataset.
The only two mistakes, shown below, are probably due to the fact that the fingers are separated, while usually they are not in the paper hand gesture, this distance may make them look like the scissor hand gesture.





External validity

To control the actual prediction capability of the model the last neural network, I.E. the best one according to the metrics, was tested with an external homemade dataset. It is composed of 210 images, 70 for each class, and the background is white and red. No data augmentation was made for this dataset and it was normalized using the same values of the train set that were previuosly computed.
The model accuracy is 33.94% and the model always predicts the same class (paper), as it can be seen in the confusion matrix.



Thus the high accuracy for the data from the dataset and the low accuracy in the external check indicates that the model is probably learning shortcuts to correctly label the images in the dataset rather than learning the actual concept of hand posture. This is probably due to the dataset size, the homogeneous background and light that on one hand make it easier for the model to classify the images but on the other hand makes it more difficult to handle real world external images. So, despite the data augmentation, the domain shift underlines a possible spurious correlation or overfittingon the domain dataset Rock-Paper-Scissor.
To sum up the results the model is not generally able to label hand postres, buti s god to do so only in certain conditions (e.g. green background).