Visual Intelligence - 2022/2023 Project

Gabriele Roncolato, Leonardo Zecchin, Alberto Carli

2D Classification - CNN vs Scatter Transform

Project Goal

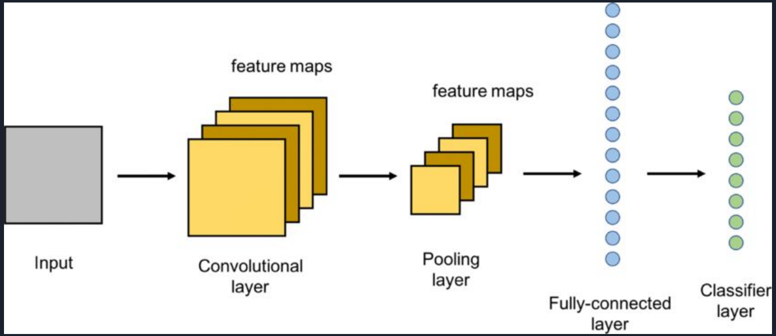
The main goal of this project is to compare the performance of a Convolutional Neural Network (CNN) architecture against the Scatter Transform in a generic 2D classification task.

In order to compare the two different techniques we have extracted the fully connected layers from the CNN and used them as a standalone Neural Network (NN) so that the classifiers of the two models are exactly the same.

This means that the only difference between these two approaches is the feature extraction phase: we are comparing the CNN convolutional layers against the Scatter Transform filters.

CNN Architecture

For the CNN we have employed a classic architecture comprised of **convolutional layers** for feature extraction followed by **fully connected layers** for classification.



We have employed multiple **dropout**, **batch norm** and **max pooling** layers to avoid overfitting.

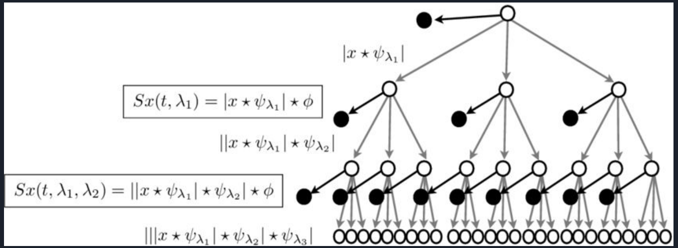
RELU is the only activation function we have used.

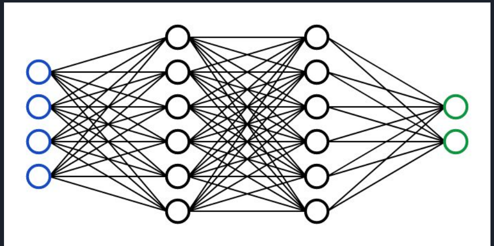
Scatter Transform Architecture

For the Scatter Transform we have designed our own architecture comprised of two main steps:

- feature extraction through the Scatter Transform
- classification through a NN comprised of fully connected and dropout layers

In particular, we have maintained the same fully connected layers as those present in the CNN architecture to achieve a more precise comparison between the two.





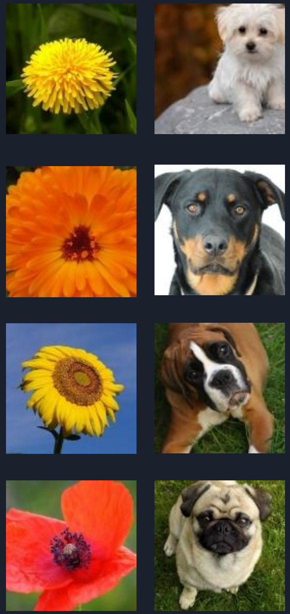
Dataset

We have used a dataset consisting of 128×128 RGB images split into two subjects:

- **dogs**;
- **flowers**.

There are 1600 pictures of dogs and 1387 pictures of flowers.

In order to make the task less trivial we have converted all of the samples to **grayscale**.



Dataset

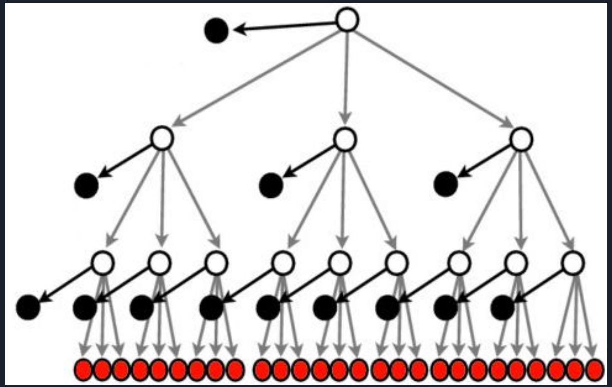
Although the problem might seem trivial, we have noticed some images where the separation between the two classes is not clear, especially when working in grayscale.

Scatter Transform Classification

We have implemented a scatter transform pipeline in MATLAB and then fed the extracted features to the NN.

In order to achieve a higher accuracy we have only selected the **highest order scatter coefficients** which, in our case, correspond to the second scatter level coefficients.

The picture on the right shows an example of which features we have selected (marked in red).



CNN Classification

Despite obtaining some initial promising results, we had noticed that the model was overifitting based on the training/validation epochs graph. In addition to that, we were not obtaining any meaningful filters in the first convolutional layer.

We have then integrated many techniques to fight overfitting such as:

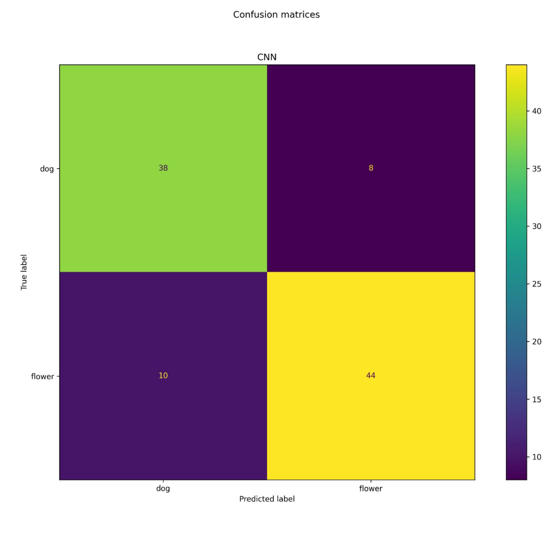
1. Batchnorm
2. Dropout
3. Data Augmentation

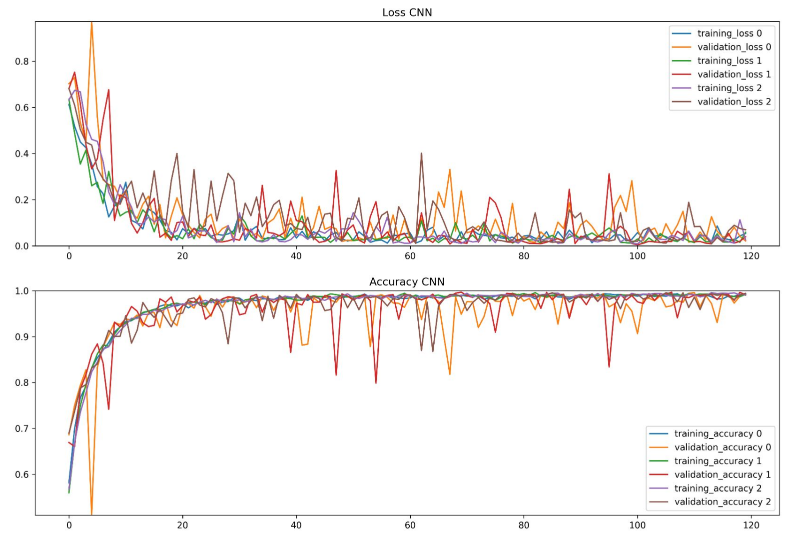
Although the accuracy and the training/validation graph of our model had improved, we could see that the filters generated by the CNN through data augmentation had numerous black bands. After running some tests, we have come to understand that these bands were caused by image translations as part of our data augmentation policy, which resulted in the CNN learning incorrectly.

To improve our results, we have therefore created a custom augmentation policy in which we only apply rotations (between -45 and +45 degrees) to the images: this improvement both increased the accuracy of our model and improved the filters in the first convolutional layer.

Best results - CNN

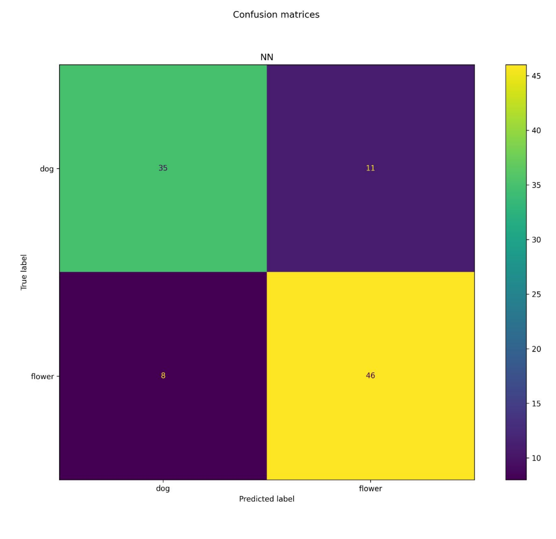
These are the best results obtained with the CNN while using 16 augmentations and the best parameters we found for the model

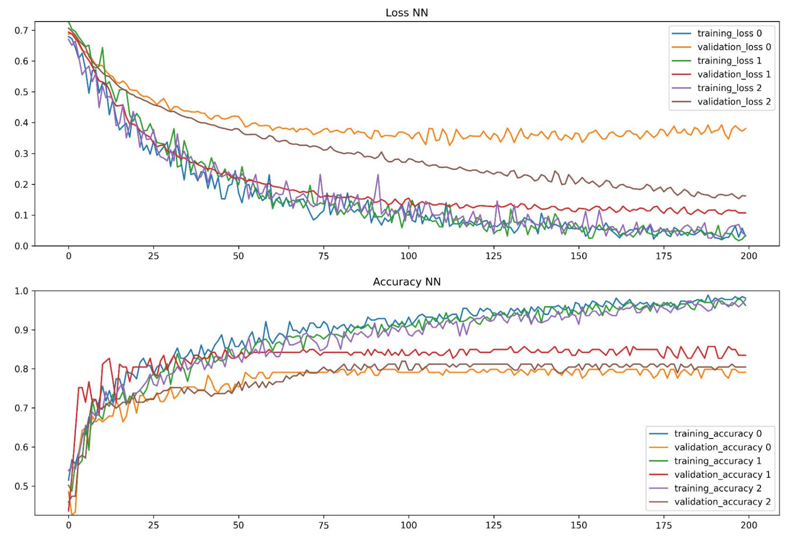




Best results - NN

These are the best results obtained with the NN model (Scatter transform + NN) while not using any augmentation and the best parameters we found for the model





CNN Filters VS Scatter Filters

The Scatter Filters are built and applied based on a fixed set of parameters: number of rotations, quality, scale and number of levels.

The CNN Filters are instead computed by the model during training, and as we have been able to observe these filters tend to look the same as the scatter ones given a generalised and large enough dataset.

Our conclusion is that the Scatter Filters are highly generic edge detection filters and that given a generalised dataset the best filters a CNN can compute are generic edge detection filters.

We have also observed that our custom policy comprised of rotations led the CNN to develop a set of filters suitable for rotated images faster.

Conclusions

Our study has revealed several key findings regarding the impact of parameter selection and data augmentation techniques on the performance of CNNs and NNs. The optimal parameters varies between the two models, but our results suggest that rotations were the most effective parameter for improving CNN's performance on our particular dataset.

Our findings highlight the importance of careful parameters selection and the potential benefits of using rotations as a major data augmentation technique for improving the performance and generalization of CNNs.

One additional finding we would like to add is that the Scatter Transform appears to give better results compared to CNNs when applied to very small datasets (less than 200-250 samples) given that the significant features can be extracted using generic edge detection filters.