Artificial Neural Networks and Deep Learning
Homework 1

Nicola Dean	Marco Fasanella	Raffaello Fornasiere	Christian Spano
10617541	10617541	10617541	10617541
nicola.dean @mail.polimi.it	marco.fasanella @mail.polimi.it	raffaello.fornasiere @mail.polimi.it	christian.spano @mail.polimi.it

1 Introduction

In order to touch all the important aspects of the procedure of finding the best solution to this classification problem, we started from our own self-made model to more sophisticated methodologies.
We can summarize our approaches in:

- Vanilla network
- Transfer Learning and Fine Tuning
- Ensemble method

Furthermore in all the attempts made, we used two classes, specifically **created to automate and support the model creation procedure**:

- Dataset Helper
- Model Helper

Through the continuous attempts, and support of methods in the two helper classes, we managed to find our best model and reach 0.8691 accuracy in the competiton.

2 Dataset Helper and Model Helper

–CHRISTIAN– Spiegazione delle due classi: lista delle funzioni e automatizzazione

3 First try: vanilla network

–NICOLA– -IMG della rete (dal lab) (magari orizzontale) risultati considerazioni

3.1 Batch Normalization

A first attempt was also adding a Batch Normalization + Relu Activation Layer before our Pooling layers. This lead to poor result due to the fact that the network was too small.

3.2 Our homemade CNN

So deepening the CNN gave this solution much significant improvements.
METTERE IMMAGINE CNN FINALE — RAFFAELLO—

3.3 Considerations

Best result consideration and observations

4 Transfer Learning and Fine Tuning

We then noticed that we needed a big change on the approach to use, because the homemade CNN was performant, but not enough. So **we started using transfer learning and Fine tuning**

4.1 Approach: Freezing Layers

The *modus operandi* that will be used from now is: freezing all layers while training on our augmented dataset the keras.application CNN, and then unlock a small amount of layers to the second phase of training (fine tuning) as near as possible to the output.

4.2 VGG19

VGG-19 is a convolutional neural network that is 19 layers deep, so we kept the freezed layers in the range of the first 8-14.
Different Data Augmentations (with different seeds and augmentation parameters) were performed between the two phases, to even increase randomisation in the two training processes.

4.2.1 Results

Freezed Layers	Accuracy	Precision	Recall	F1
8	0.8169	0.7989	0.7651	0.763
9	0.8225	0.8181	0.7682	0.7776
10	0.8338	0.8161	0.7929	0.8001
11	0.7577	0.7109	0.715	0.7048
12	0.7944	0.766	0.7504	0.7489
13	0.8028	0.7806	0.754	0.7596

Table 1: Results with Transfer Learning and Number of freezed layers for VGG19.

4.2.2 Considerations

One of the most particular observation that we can made after experimenting the first attempts of freezing, is that freezing the net until the pooling and not between convolutions leads to a better accuracy.

4.3 VGG16

–CHRISTIAN– Spiegazione modell + prove fatte

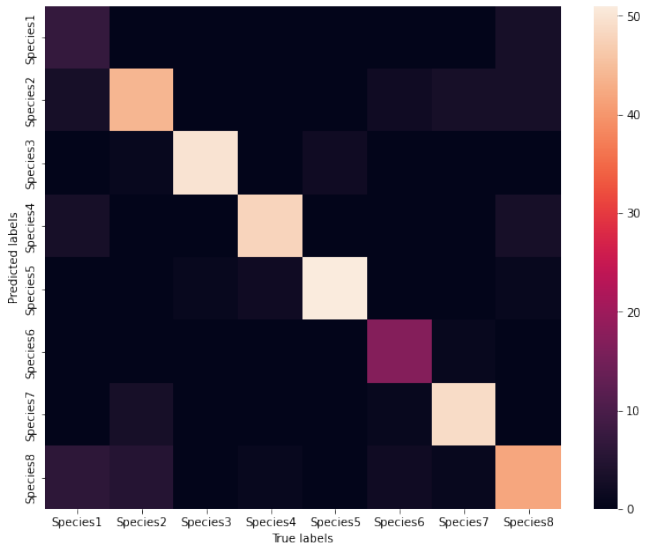

Figure 1: Confusion Matrix of best configuration with VGG19.

4.3.1 Results

4.4 Xception

4.5 Other Models

4.5.1 Resnet

–NICOLA–

4.5.2 GoogleNet

4.6 EfficientNet

–NICOLA–

5 Ensemble

–NICOLA– Approccio provato a mischiare modelli c'era bias perchè avevano seed diversi

5.0.1 Results

6 If we had more time..

con più tempo cosa avremmo provato

7 Our Submissions

Description	Result
VGG19 - 10 Freezed Layers b	0.823015873 0.8225

Table 2: Results with Transfer Learning and Number of freezed layers for VGG19.

8 Conclusions

Considerazioni finali e best model fattoo