DEEP LEARNING FOR GALAXY PHOTOMETRIC REDSHIFTS

Théophile Noble[a]

a. Mines Paris - PSL, theophile.noble@etu.minesparis.psl.eu

Keywords : galaxies; cosmology; photometric redshifts ; Deep Learning

Abstract :
In this work, we tested and optimized the CNN from Pasquet et al., 2019, in the context of the Large Synoptic Survey Telescope (LSST), to estimate the redshifts of more than 600.000 galaxies from the Sloan Digital Sky Survey (SDSS). The data are composed of 64×64 *pixels ugriz images centered on the galaxies. We looked for a tradeoff between overfitting and underfitting by investigating the influence of the inception modules, of the number of dense layers and of the number and the position of dropout layers. Most of the results do not lead to clear conclusions, since the combination of the different influences is complex to modelize. We reached a bias of* $0.4.10^{-3}$*, an outliers rate of* 0.81% *and a MAD deviation* σ_{MAD} *of* $1.19.10^{-2}$*, outperforming the results of the previous intern (results available here). More tests must be conducted on the LSST depth to investigate the dependence of the model on blended galaxies.*

1 Introduction

In the context of imaging surveys such as LSST (Large Synoptic Survey Telescope) or Euclid, a huge amount of data will be available, with unmatched area coverage, wavelength range and depth (Newman & Gruen, 2022). Automating complex tasks will then become necessary, in particular concerning the redshifts of the galaxies (D'Isanto & Polsterer, 2018). As a reminder, the redshift of an object is a shift towards the red end of its spectrum due to gravitational effects. They are in particular used to compute the distance of extra-galactic sources and study the large-scale structure of the universe.

The spectroscopic redshifts are derived using the measure of the electromagnetic spectral energy distribution (SED). Due to the expansion of the universe, the SED of an object is stretched towards longer wavelengths with a factor $1 + z$, with z the redshift. The identification in the SED of known features enables to determine the amount it has been stretched, and so to derive the redshift (Salvato et al., 2018). On the other hand, photometric redshifts are derived using the measure of the flux of an object through multiple filters, each of which measuring the brightness of the object in a certain range of wavelength (Newman & Gruen, 2022). In some way, it is like taking a picture of the objects, with other and broader filters than the classical RGB. Spectroscopic redshifts are very precise but also time consuming. Hence, the use of photometric redshifts, less accurate but less time-intensive, has become necessarly (D'Isanto & Polsterer, 2018).

To derive photometric redshifts, three main solutions exist: template-fitting (where the fluxes measured in each of the filters are compared with existing templates of the galaxy SED), traditional Machine Learning methods (less time-intensive but still requiring a manually extraction of the features from the images in each filter) and Deep Learning methods, that recently succeded to derive photometric redshifts directly from multi-band images (Pasquet et al., 2019), making the feature extraction obsolete and outperforming the previous techniques (D'Isanto & Polsterer, 2018).

Deep Learning models are based on artificial neural networks, inspired by how the human brain learn information. Its purpose is to learn a function from data, to be capable of predicting the output from an input. An artificial neuron takes a matrix of inputs, combines them linearly by multiplying the inputs by a matrix of weights and summing it to obtain a single value, and applies an activation function to

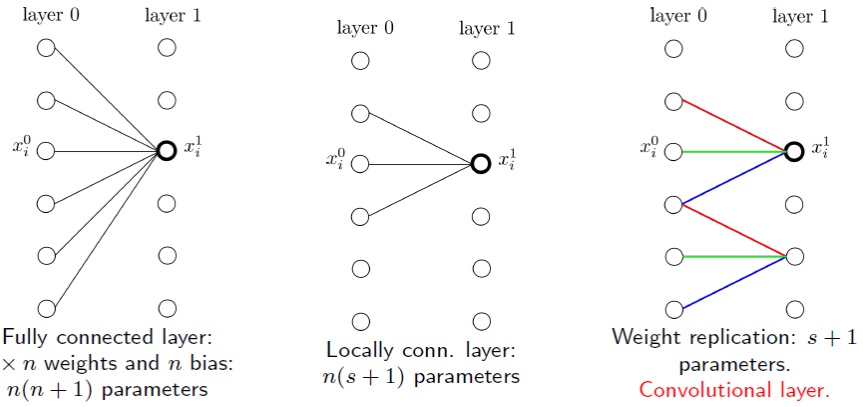


Figure 1: Explanation of convolutional layer from the course of E. Decencière (Mines Paris - PSL, 2023)

the result, that decides if information is transmitted through the neuron or not. In a neural network, the neurons are organized in layers, the inputs of each layer being the outputs of the previous layer. The inputs of the first layer are for instance the pixels of an image. The depth of a network is defined as the number of layers in the network.

A loss function is chosen depending on the task. The loss function represents in some way the difference between the prediction and the true value (also called the target). The principle of a neuron network is so to minimize this loss function during the training, thanks to the data. Generally, the input data are split in three datasets: the training set (generally 64% of the data), the validation set (generally 16% of the data) and the test set (generally 20% of the data). In supervised Deep Learning, the data have labels that are the true values.

The network is trained during a chosen number of epochs. During each epoch, the model sees the entire training dataset and updates the weights to minimize the loss function. It also sees the validation dataset to evaluate its performances and find hyperparameters. It is the learning step. When the training is ended, the model is evaluated on the test set, i.e. it is fed with data it has never seen, without any label. Then, the results are compared to the labels to evaluate the performances of the final model.

If each neuron of a layer is connected to all the neurons of the previous layer, it is called a fully connected layer. A convolutional layer is a fully connected layer where the weights of non-local connections are set to zero (local structure) and all the neurons of a same layer share the same weights (translation invariance). The number of neurons to which each neuron of the layer is connected is called the kernel of the layer. For instance, 5×5 means that each neuron of the layer is connected to the neurons of the previous layer contained in a square of area 5×5. This enables to shrink the number of parameters and make the network more manageable. One can see Figure 1 for a better understanding.

Convolutional neural networks (hereafter CNN) are feed-forward artificial neural networks containing several iterations of convolutional layers with increasing depth, followed by optionnal layers such as Average Pooling (a subsampling layer where the information contained in a certain number of neurons is averaged in a single neuron), and finally fully connected layers (or Dense layers) known for extracting the meaningful features of the data.

In particular, the model proposed in Pasquet et al., 2019, is a convolutional neural network. The architecture of the model can be found on Figure 4a, where Conv2D stands for convolutional layer. The model follows the classical pattern of CNN. The inception module, introduced in Szegedy et al., 2014, is composed of four parallel branches, containing respectively: a Conv2D with kernel 1×1 and a Conv2D with kernel 3×3; a Conv2D with kernel 1×1 and a Conv2D with kernel 5×5; a Conv2D with kernel 1×1 and an average pooling layer; and a Conv2D with kernel 1×1. Its architecture can be seen on Figure 2a. Before the fully connected part, the reddening of the galaxies is added. The output of the

network has been modified: it was a probability density function on the original paper, it is now directly the redshift of the galaxy.

The purpose of this work is to test and optimize the CNN of Pasquet et al., 2019, in the context of the LSST collaboration.

In Section 2, we describe the different models tested, the data we used and the metrics we computed to compare the models. The results of these models are presented in Section 3; we investigated the influence of the inception modules and of different layers on the performances of the models. In Section 4, we discuss the influence of the inception modules and of the different layers we tested. The results are summarized in Section 5.

2 Methodology

2.1 CNN

To optimize the model, we have to find a tradeoff between a too complex model and a too simple model. We assume that a too complex model will overfit because it will learn too complex features from the training data. Overfitting is a common issue encountered in Deep Learning, meaning that the model will learn too much specific caracteristics of the training dataset. Therefore, the training error will be very low and there will be a large gap between training and validation errors. On the other hand, a too simple model will underfit, since it will not be able to learn enough features from the training data. To investigate this problem, we will first focus on two simplified versions of Pasquet and then focus on the relation between the number of dense layers and the performances of the model, since the dense layers extract the main specific features of the images.

To fight overfitting, we will also use a Learning Rate Scheduler (meaning that the learning rate depends on the epoch) and an early stopping (meaning that the training stops earlier if the validation error increases). We will also use Dropout to avoid memorization. Dropout is a technique where each neuron of a specific layer has a certain probability (generally 0.5) of being present in the model (Srivastava et al., 2014). An other effect of the Dropout layers is to increase the robustness of the models (Gal & Ghahramani, 2016). We will investigate the relation between the position and the numbers of dropout layers and the performances of the model.

We made the hypothesis that the two sequences AveragePooling - Inception module - Inception module play a similar role in the extraction of features. Hence, we decided to remove two of the five inception modules in a model that we called Pasquet3i.

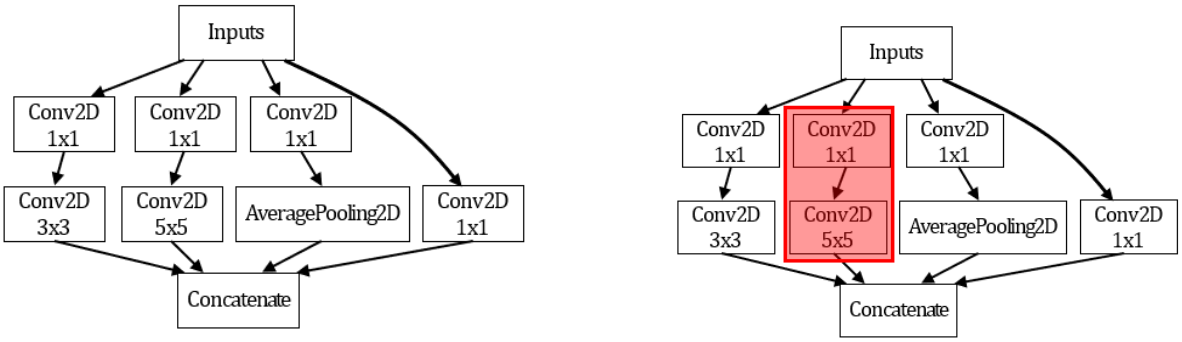
We also assumed that the two branches with two convolutional layers in the inception module play a similar role. Hence, we decided to also remove one of these two branches in a model that we called Pasquet-simplf.

Our work is organised in 3 branches that correspond to the 3 different baseline models[1]:

- Pasquet: The exact CNN from J.Pasquet et al. (2019).
- Pasquet3i: The modified version in which we removed 2 of the 5 inception modules. The architecture of Pasquet3i can be found on Figure 4b.
- Pasquet-simplf: The modified version in which we removed 2 of the 5 inception modules, the dense layer of shape 256 and one of the four parallel branches in the inception module itself. The architecture of Pasquet-simplf can be found on Figure 4c. The architecture of the inception module can be found on Figure 2b.

Each of the branches follows the same pattern in variation of the number of dense layers and dropout layers. The number of dense layers goes from 2 to 5, from 2 to 4 and from 2 to 3 for Pasquet, Pasquet3i

[1]To run the CNN, we used the DANTE cluster provided by the IPGP.



(a) Architecture of the inception module used in Pasquet and Pasquet3i

(b) Architecture of the inception module used in Pasquet-simplf

Figure 2: Architecture of the inception modules

and Pasquet-simplf, respectively. The number of dropout layer is positive and inferior or equal to the number of dense layers. For certain models, several positions of the dropout layers will be tested. This is summarized in the first five columns of Table 7.

Each of the models has a name based on the branch it belongs to (Pasquet, Pasquet3i or Pasquet-s), the number of dropout layers it contains (abbreviated as "dr") and the number of dense layers it contains (abbreviated as "dn"). When the position of the dropout layers is ambiguous, it is precised at the end of the name. In most of the cases, the dropout layers will be inserted between each dense layer. "beg" means that the dropout layers are inserted from the top of the fully connected part of the model. "end" means that the dropout layers are inserted from the bottom of the fully connected part. When a dropout layer is inserted elsewhere, it is explicitly noted.

For instance, a model based on Pasquet with 2 dropout layers and 4 dense layers where the dropout layers are inserted from the bottom of the fully connected part of the model (which means that the end of the model is the sequence: Dense - Dense - Dropout - Dense - Dropout - Dense) will be called "Pasquet2dr4dnend". Thus, the baselines can be named with this method: Pasquet would be called Pasquet0dr3dn, Pasquet3i would be called Pasquet3i0dr3dn and Pasquet-simplf would be called Pasquet-s0dr2dn.

The models for which at least one of the metrics (defined in the metrics section) is inferior to the metrics of Pasquet will then be tested with data augmentation. Data augmentation is a common technique used in Deep Learning that consist in applying random transformations to the data in order to obtain more diversity in the images of the dataset. Here, we will use random rotations and random flips. We cannot use random translations nor random crops, since the galaxies should always remain integrally inside the images and we do not know which area of the images the galaxies cover.

2.2 Data

We used the same data as Pasquet et al., 2019. The dataset is composed of 659.857 images of the Sloan Digital Sky Survey (SDSS) galaxies (York et al., 2000). The images have a size of 64×64 pixels and contains the 5 filters u, g, r, i and z. The galaxies are in the center of the images. For each image, we had access to the spectroscopic redshift (z with $z \leq 0.6$) and the reddening (ebv) of the galaxies. The data are public and can be found here.

2.3 Metrics

To be able to compare our results with the results of Pasquet et al., 2019, we used the 3 main metrics presented in the paper. They are based on the residuals, that mesure the relative difference between the

photometric redshifts and the spectroscopic redshifts: $\Delta z = \frac{z_{photo} - z_{spec}}{1+z_{spec}}$ where z_{photo} is the predicted photometric redshift and z_{spec} the given spectroscopic redshift. The closer the residuals are to 0, the better the model performs. The metrics are:

- The predicted bias which is the mean of the residuals. It gives an overall view of the accuracy of the model.
- The fraction of outliers η which is the fraction of images for which the residuals are greater than 0.05 in absolute value: $|\Delta z| = |z_{spec} - z_{photo}| > 0.05$. If the number of outliers decreases, that means that there are more predictions closer to the spectroscopic values but it does not give any information about the variability of the residuals in the accepted band.
- The MAD Deviation $\sigma_{\text{MAD}} = 1.4826 \times \text{MAD}$ where $\text{MAD} = \text{Median}\left(|\Delta z - \text{Median}(\Delta z)|\right)$. It measures the variability of the predictions, with little dependence on the outliers.

The three metrics have to be considered together in order to have a correct view of the accuracy of the model. Therefore, a model can be told better than another one if all its metrics are smaller.

3 Results

In this section, we present the results of the tests of the different CNNs derived from each baseline (Pasquet, Pasquet3i and Pasquet-simplf).

The models we tested to investigate each question are not all displayed in the following sections but we chose models that represented the global results. To have an overview of all the models that were tested, one can look on Table 7.

It is interesting to keep in mind, for the purpose of optimizing Pasquet, that even if a model has poorer performances than another model which we compare to, it can still perform better than Pasquet.

Figure 3 shows the results for the 3 baselines (Pasquet, Pasquet3i and Pasquet-simplf). Each point is a redshift prediction. In absissa there are the spectroscopic redshifts (the targets) and in ordinate there are the photometric reshift (the predictions). Therefore, the closer the points are to the bisector, the better the model performs.The concentration of points in a certain zone is represented by a color scale. We have similar figures for each of the tested model but they are not shown here, since it does not bring much information.

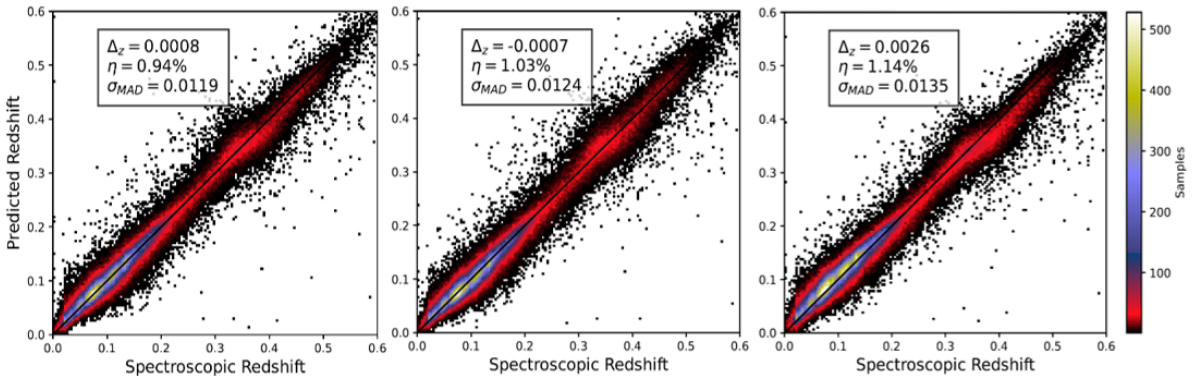


Figure 3: Comparison of the 3 baselines (from left to right: Pasquet, Pasquet3i and Pasquet-simplf)

3.1 Influence of the inception modules

In order to study the influence of the inception modules in the performances of the model, we tested several architectures for models derived from the three baselines, presented in 1.

If we look at the bias and the outliers rate, we can see that no clear conclusion can be drawn, except that it seems that the models based on Pasquet-simplf perform less well than the others.

Focusing on the MAD deviation, we see that the models derived from Pasquet almost always have a smaller devation than the models based on Pasquet3i or Pasquet-simplf, except for Pasquet3i2dr4dn that has a smaller MAD deviation than the corresponding model based on Pasquet. In contrary, there is no clear ranking between the models based on Pasquet3i and the models based on Pasquet-simplf, the MAD deviation being close one to another.

The results suggest that oversimplifying the models (as we did in Pasquet-simplf) leads to poorer performances. On the contrary, removing two inception modules and one average pooling layer (as we did in Pasquet3i) can deteriorate the performances or improve them. It seems therefore that the ranking that can be done between the models looking at their performances is not only based on the number or on the architecture of inception modules, but that it would be a sort of competition or combination between the different parameters (including dropout and dense layers).

CNN	Number of dropout layers	Number of dense layers	Number of inception modules	Δz (10^{-3})	η	σ_{MAD} (10^{-2})
Pasquet0dr2dn	0	2	5	1.0	0.98%	1.21
Pasquet3i0dr2dn	0	2	3	0.8	0.95%	1.24
Pasquet-simplf	0	2	3	2.6	1.14%	1.35
Pasquet1dr2dn	1	2	5	-1.3	0.90%	1.20
Pasquet3i1dr2dn	1	2	3	1.0	0.85%	1.23
Pasquet-s1dr2dn	1	2	3	1.7	0.94%	1.22
Pasquet	0	3	5	0.8	0.94%	1.19
Pasquet3i	0	3	3	-0.7	1.03%	1.24
Pasquet-s0dr3dn	0	3	3	0.4	1.08%	1.25
Pasquet1dr3dnend	1	3	5	-1.9	0.92%	1.19
Pasquet3i1dr3dnend	1	3	3	-0.7	0.9%	1.26
Pasquet-s1dr3dnend	1	3	3	0.5	1.0%	1.25
Pasquet2dr3dn	2	3	5	7.2	1.03%	1.36
Pasquet3i2dr3dn	2	3	3	11.3	2.0%	1.73
Pasquet-s2dr3dn	2	3	3	11.9	2.79%	1.99
Pasquet1dr4dnbeg	1	4	5	11.4	2.21%	1.54
Pasquet3i1dr4dnbeg	1	4	3	16.2	3.82%	1.63
Pasquet2dr4dnend	2	4	5	6.6	1.1%	1.35
Pasquet3i2dr4dnend	2	4	3	4.5	1.02%	1.29

Table 1: Comparison of the results for different numbers of inception modules and different architectures of inception module

3.2 Influence of the number of dense layers

To highlight the influence of the number of dense layers on the performances, independently of dropout layers, we tested different models without dropout layers. The results are shown on Figure 2.

Focusing on Pasquet, one can see that the model with two dense layers (Pasquet0dr2dn) has poorer performances than the model with three dense layers (Pasquet0dr3dn). It is the same for the models based on Pasquet-simplf. So, it seems that the accuracy of the model increases with the number of dense

layers. However, the trend is reversed for Pasquet3i. In this case, the performance decreases with the number of dense layers.

Therefore, the results suggest that the influence of the dense layers depends on the architecture of the model.

CNN	Number of dropout layers	Number of dense layers	Number of inception modules	Δz (10^{-3})	η	σ_{MAD} (10^{-2})
Pasquet0dr2dn	0	2	5	1.0	0.98%	1.21
Pasquet	0	3	5	0.8	0.94%	1.19
Pasquet3i0dr2dn	0	2	3	0.8	0.95%	1.24
Pasquet3i	0	3	3	-0.7	1.03%	1.24
Pasquet3i0dr4dn	0	4	3	1.8	1.08%	1.26
Pasquet-simplf	0	2	3	2.6	1.14%	1.35
Pasquet-s0dr3dn	0	3	3	0.4	1.08%	1.25

Table 2: Comparison of the results for different numbers of dense layers

3.3 Influence of the position of the dropout layers

To investigate the influence of the position of the dropout layers, we tested two types of positions: on one hand, we placed the dropout layers among the dense layers at the end of the model, and on the other hand, we placed a dropout layer among the inception modules. The results are presented on Figure 3 and 4, respectively.

For the first part of the study, we tested different models with the same number of dropout layers and the same number of dense layers, based on Pasquet and Pasquet3i. One can see that, for instance, Pasquet1dr4dnbeg (which end is the sequence: Dense - Dropout - Dense - Dense - Dense) performs less well than Pasquet1dr4dnend (which end is the sequence: Dense - Dense - Dense - Dropout - Dense). We have the same kind of results for Pasquet3i1dr3dnbeg and Pasquet3i1dr3dnend, that also shows that the performances can be much deteriorated by placing the dropout layer in the beginning of the fully connected part. However, for Pasquet2dr5dnbeg and Pasquet2de5dnend, the MAD deviation is smaller for Pasquet2dr5dnbeg than for Pasquet2dr5dnend.

Therefore, the results suggest that placing one or several dropout layer from the top of the fully connected part of the model leads to poorer performances than placing it or them from the bottom of the fully connected part. However, we also tested Pasquet3i2dr4dn (1-2/3-4) where the dropout layers where placed between the first and the second dense layers and the third and the fourth dense layers. This model gives better performances than Pasquet3i2dr4dnend for all the metrics. The conclusion appears to only apply to models where the dropout layers are positioned to fill the spaces between the dense layers.

For the second part of the study, we tested models where there were already dropout layers between each pair of dense layers. In Figure 4, the part in parenthesis indicates the position of the additional dropout: "i" is for "inception", it corresponds to the inception module after which the dropout layer was inserted. Hence, "1dr after 4^{th}" means that we put one dropout layer after the fourth inception module. For every model, this operation leads to poorer performances except for Pasquet2dr3dn, which bias is greater than the bias of Pasquet3dr3dn with one dropout after the fourth inception module. So, the results suggest that adding a dropout layer among inception modules leads to poorer performances.

Therefore, the results suggest that the position of the dropout layers plays a major role in the performances of the models. It seems that greater performances can be obtained when the only dropout layers added are inserted from the bottom of the fully connected part at the end of the models, or when they are inserted from the top and the bottom on the same time.

CNN	Number of dropout layers	Number of dense layers	Number of inception modules	Δz (10^{-3})	η	σ_{MAD} (10^{-2})
Pasquet1dr4dnbeg	1	4	5	11.4	2.21%	1.54
Pasquet1dr4dnend	1	4	5	0.4	0.87%	1.22
Pasquet1dr5dnbeg	1	5	5	1.24	1,87%	1.46
Pasquet1dr5dnend	1	5	5	1.1	0,86%	1.21
Pasquet2dr5dnbeg	2	5	5	5.8	1.14%	1.39
Pasquet2dr5dnend	2	5	5	3.9	0.99%	1.42
Pasquet3i1dr3dnbeg	1	3	3	-2.6	12.56%	2.98
Pasquet3i1dr3dnend	1	3	3	-0.7	0.9%	1.26
Pasquet3i1dr4dnbeg	1	4	3	16.2	3.82%	1.63
Pasquet3i1dr4dnend	1	4	3	-1.9	0.95%	1.25
Pasquet3i2dr4dn (1-2/3-4)	2	4	3	3.0	0.89%	1.25
Pasquet3i2dr4dnend	2	4	3	4.5	1.02%	1.29

Table 3: Comparison of the results for different positions of dropout layers among the dense layers

CNN	Number of dropout layers	Number of dense layers	Number of inception modules	Δz (10^{-3})	η	σ_{MAD} (10^{-2})
Pasquet2dr3dn	2	3	5	7.2	1.03%	1.36
Pasquet3dr3dn (1dr after 4^{th} i)	3	3	5	6.4	1.1%	1.43
Pasquet3i2dr3dn	2	3	3	11.3	2.0%	1.73
Pasquet3i3dr3dn (1dr after 2^{nd} i)	3	3	3	16.4	5.25%	1.96
Pasquet-s1dr2dn	1	2	3	1.7	0.94%	1.22
Pasquet-s2dr2dn (1dr after 2^{nd} i)	2	2	3	5.0	1.35%	1.31

Table 4: Comparison of the results for positioning a dropout layer among the inception modules

3.4 Influence of the number of dropout layers

To investigate the influence of the number of dropout layers on the performances of the models, we tested different models based on the three baselines, for different numbers of dense layers and different number of dropout layers. The results are presented on Figure 5.

One can see that Pasquet1dr3dnend has a bias superior to Pasquet0dr3dn, but an outliers rate smaller and a MAD deviation equal. Pasquet3i1dr3dnend and Pasquet-s1dr3dnend has also a smaller outliers rate than the corresponding model with no dropout layer but the effect on the other metrics is not clear. For the models with more than one dropout layer presented here, the metrics are always greater than the corresponding model with one dropout layer.

Once again, the results do not highlight any clear relation between the number of dropout layers and the performances of the models. It suggests that adding one dropout layer improves the outliers rate but the effect on the bias and MAD deviation is unclear. It seems that the MAD deviation is not much modified. On the contrary, adding more than one dropout seems to deteriorate the performances with respect to the model with one dropout layer. In this case, it seems that the combination of the effects of the dense layers and of the dropout layers do not allowed to highlight the influence of one or the other.

CNN	Number of dropout layers	Number of dense layers	Number of inception modules	Δz (10^{-3})	η	σ_{MAD} (10^{-2})
Pasquet0dr2dn	0	2	5	1.0	0.98%	1.21
Pasquet1dr2dn	1	2	5	-1.3	0.90%	1.20
Pasquet	0	3	5	0.8	0.94%	1.19
Pasquet1dr3dnend	1	3	5	-1.9	0.92%	1.19
Pasquet2dr3dn	2	3	5	7.2	1.03%	1.36
Pasquet3dr3dn (1dr after 4^{th})i	3	3	5	6.4	1.1%	1.43
Pasquet3i	0	3	3	-0.7	1.03%	1.24
Pasquet3i1dr3dnbeg	1	3	3	-2.6	12.56%	2.98
Pasquet3i1dr3dnend	1	3	3	-0.7	0.9%	1.26
Pasquet3i2dr3dn	2	3	3	11.3	2.0%	1.73
Pasquet3i3dr3dn (1dr after 2^{nd})i	3	3	5	16.4	5.25%	1.96
Pasquet-s0dr3dn	0	3	3	0.4	1.08%	1.25
Pasquet-s1dr3dnend	1	3	3	0.5	1.0%	1.25
Pasquet-s2dr3dn	2	3	3	11.9	2.79%	1.99

Table 5: Comparison of the results for different numbers of dropout layers

3.5 Influence of data augmentation

In order to optimize Pasquet, we added data augmentation to the models for which at least one metric was better than Pasquet. The results are presented on Figure 6.

One can see that, for instance, adding data augmentation to Pasquet3dr4dn leads to poorer performances for all the metrics. On the contrary, for Pasquet3i1dr2dn, the performances are improved by the data augmentation. Otherwise, Pasquet-s1dr3dn has a greater bias with data augmentation, but smaller outliers rate and MAD deviation σ_{MAD}.

Hence, the results do not lead to a clear conclusion. The overall effect of data augmentation is to improve the outliers rate. In most of the cases, it degrades the bias. There is no trend in the variation of the MAD deviation σ_{MAD}.

4 Discussion

The variation of the performances between the different branches can be explained by looking at overfitting and underfitting. Since reducing the number of inception modules reduces the complexity of the module, in most of the cases it should also reduces the overfitting of the model, leading generally to a greater training rate, what can degrade the performances or not depending on the effect on the validation error, hence the uncertainty in the ranking between Pasquet and Pasquet3i.

In the same time, the simplified models seems to be not complex enough to sufficiently learn the characteristics of the images, leading to underfitting. This can explain why the models based on Pasquet-simplf perform less well than the others.

Adding dense layers is a way to increase the complexity of the model and fight underfitting. On the contrary, adding dropout layers is a way to fight overfitting by reducing the capacity of the model to memorize. Hence, these two operations have competitive effects on the models. This explains the difficulty to isolate the effects of one or the others.

For the same reason, adding one dense layer can contribute to allow the model to better extract the meaningful features. That is especially the case for Pasquet-simplf because of the over-simplification of

CNN		Number of dropout layers	Number of dense layers	Number of inception modules	Δz (10^{-3})	η	σ_{MAD} (10^{-2})
Pasquet	Pasquet1dr2dn	1	2	5	1.9	0.82%	1.20
	Pasquet0dr3dn	0	3	5	1.0	0.86%	1.16
	Pasquet1dr3dnend	1	3	5	-0.4	0.87%	1.21
	Pasquet1dr4dnend	1	4	5	0.7	0.86%	1.19
	Pasquet3dr4dn	3	4	5	7.4	1.22%	1.46
	Pasquet1dr5dnend	1	5	5	0.7	0.91%	1.24
Pasquet3i	Pasquet3i0dr2dn	0	2	3	1.8	1.05%	1.26
	Pasquet3i1dr2dn	1	2	3	0.4	0.81%	1.19
	Pasquet3i0dr3dn	0	3	3	1.7	1.07%	1.23
	Pasquet3i1dr3dnend	1	3	3	-0.6	0.87%	1.24
	Pasquet3i2dr4dn (1-2/3-4)	2	4	3	3.0	0.85%	1.21
Pasquet-simplf	Pasquet-s1dr2dn	1	2	3	0.1	0.88%	1.25
	Pasquet-s0dr3dn	0	3	3	-1.1	1.04%	1.23
	Pasquet-s1dr3dn	1	3	3	-0.9	0.94%	1.24

Table 6: Results of the tests using data augmentation, grouped by the baseline they belong to and sorted in ascending number of dense layers and number of dropout layers.

the models. On the contrary, adding more dense layers can have an opposite effect, since the model will tend to learn not representative features.

The position of the dropout layers also seems to play a major role in the capacity of the model to learn the characteristics of the model. Since the dropout layer divides by two the number of useful pixels, placing them too early in the model (among the inception modules or at the top of the fully connected part) implies that the model is not able to learn sufficiently. Similarly, adding too much dropout layers increases the risks of underfitting.

The effect of data augmentation seems to be explained by the competition between two effects: it adds images in the dataset (implying a better learning) but it also increases the diversity of the images (implying that the model would have to learn more features). It is possible that with a larger dataset, the data augmentation always leads to better performances.

It finally appears that the model that seems to perform better is Pasquet3i1dr2dn: a model based on Pasquet3i (therefore with three inception modules), with one dropout layer and two dense layers. It is the better model for which the three metrics are equal or better to or than Pasquet.

5 Conclusion

We investigated different ways to optimize the CNN from Pasquet et al., 2019. We defined three baselines from which we tested several models, varying the number of dropout layers and dense layers. We applied data augmentation to the 14 models that performs better. We highlighted that the model that performs better is a model with three inception modules, one dropout layer and two dense layers. This model outperfoms the results of the previous intern, available here.

Further analyses must be conducted to verify the hypothesis that the two layers of inception modules play a similar role. In particular, one should test models with four inception modules or more than five. It will also be interesting to verify that the model is sufficiently robust, as it is said in Pasquet et al., 2019 (i.e. study the variation of the performances between two tests for a same model). Finally, the model has to be trained on a larger dataset with images at the depth of LSST, to investigate its dependence on

blending galaxies.

References

D'Isanto, A., & Polsterer, K. L. (2018). Photometric redshift estimation via deep learning - generalized and pre-classification-less, image based, fully probabilistic redshifts [Publisher: EDP Sciences]. *Astronomy & Astrophysics*, *609*, A111. https://doi.org/10.1051/0004-6361/201731326

Gal, Y., & Ghahramani, Z. (2016, October 4). Dropout as a bayesian approximation: Representing model uncertainty in deep learning [version: 6]. https://doi.org/10.48550/arXiv.1506.02142

Newman, J. A., & Gruen, D. (2022). Photometric redshifts for next-generation surveys. *Annual Review of Astronomy and Astrophysics*, *60*(1), 363–414. https://doi.org/10.1146/annurev-astro-032122-014611

Pasquet, J., Bertin, E., Treyer, M., Arnouts, S., & Fouchez, D. (2019). Photometric redshifts from SDSS images using a convolutional neural network. *Astronomy & Astrophysics*, *621*, A26. https://doi.org/10.1051/0004-6361/201833617

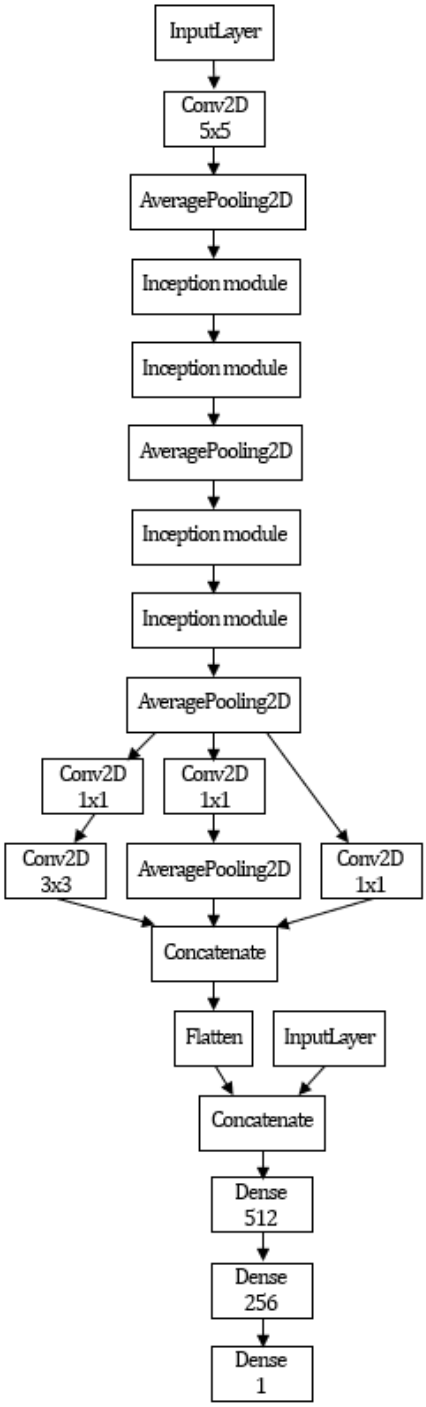
Salvato, M., Ilbert, O., & Hoyle, B. (2018, June 4). The many flavours of photometric redshifts. https://doi.org/10.48550/arXiv.1805.12574

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*(56), 1929–1958. Retrieved February 8, 2024, from http://jmlr.org/papers/v15/srivastava14a.html

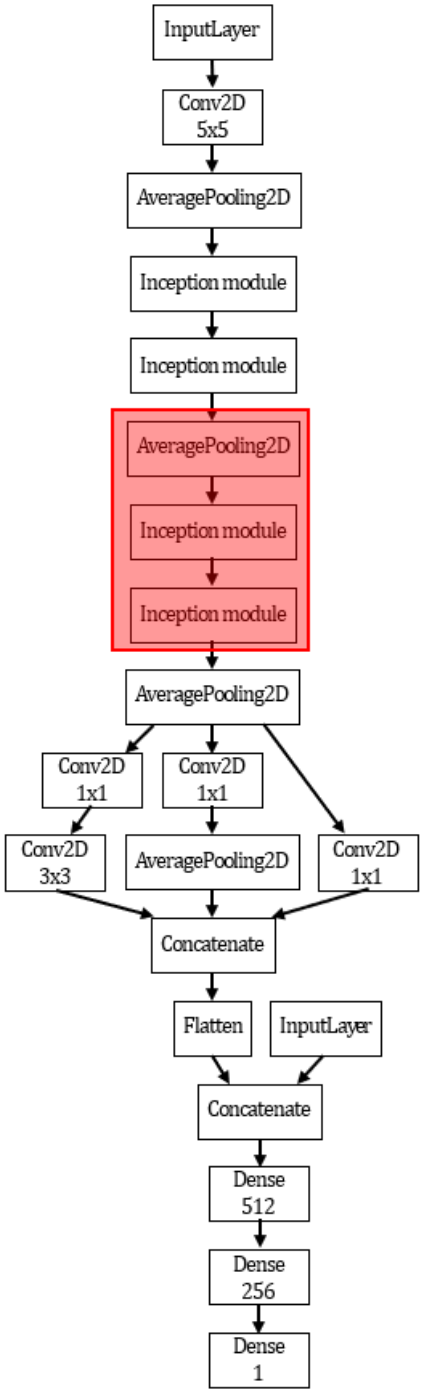
Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2014, September 16). Going deeper with convolutions. https://doi.org/10.48550/arXiv.1409.4842

York, D. G., Adelman, J., Anderson, J. E., Jr., Anderson, S. F., Annis, J., Bahcall, N. A., Bakken, J. A., Barkhouser, R., Bastian, S., Berman, E., Boroski, W. N., Bracker, S., Briegel, C., Briggs, J. W., Brinkmann, J., Brunner, R., Burles, S., Carey, L., Carr, M. A., ... SDSS Collaboration. (2000). The sloan digital sky survey: Technical summary [ADS Bibcode: 2000AJ....120.1579Y]. *The Astronomical Journal*, *120*, 1579–1587. https://doi.org/10.1086/301513

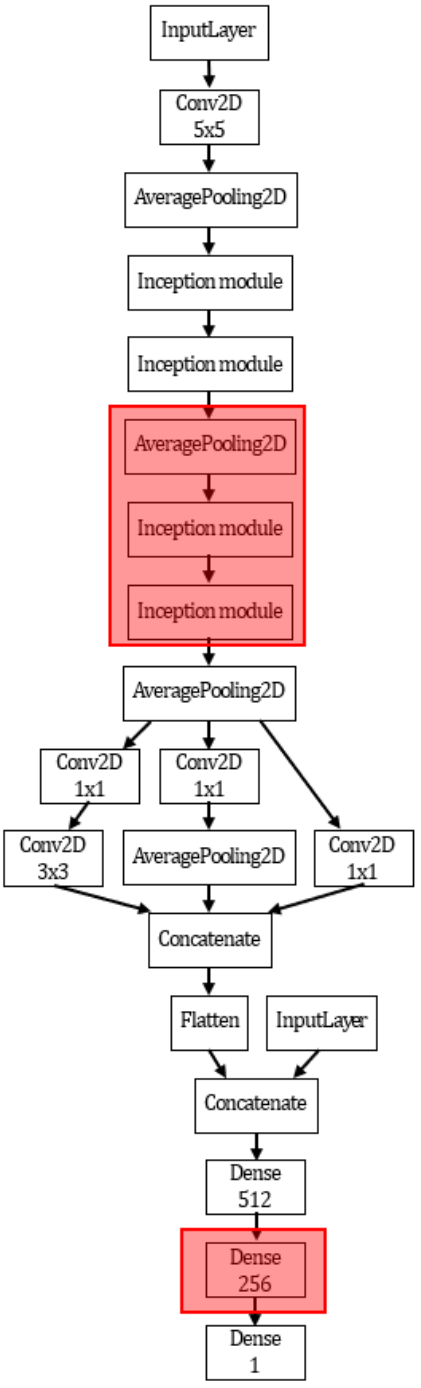
A Architectures of Pasquet, Pasquet3i and Pasquet-simplf



(a) Architecture of Pasquet



(b) Architecture of Pasquet3i



(c) Architecture of Pasquet-simpf

Figure 4: Architectures of the three baselines

B General Results

CNN		Number of dropout layers	Number of dense layers	Number of inception modules	Δz (10^{-3})	η	σ_{MAD} (10^{-2})
Pasquet	Pasquet	0	3	5	0.8	0.94%	1.19
	Pasquet0dr2dn	0	2	5	1.0	0.98%	1.21
	Pasquet1dr2dn	1	2	5	-1.3	0.90%	1.20
	Pasquet1dr3dnend	1	3	5	-1.9	0.92%	1.19
	Pasquet2dr3dn	2	3	5	7.2	1.03%	1.36
	Pasquet3dr3dn (1dr after 4$^{\text{th}}$ i)	3	3	5	6.4	1.1%	1.43
	Pasquet1dr4dnbeg	1	4	5	11.4	2.21%	1.54
	Pasquet1dr4dnend	1	4	5	0.4	0.87%	1.22
	Pasquet2dr4dnend	2	4	5	6.6	1.1%	1.35
	Pasquet3dr4dn	3	4	5	4.0	0.93%	1.28
	Pasquet1dr5dnbeg	1	5	5	1.24	1,87%	1.46
	Pasquet1dr5dnend	1	5	5	1.1	0,86%	1.21
	Pasquet2dr5dnbeg	2	5	5	5.8	1.14%	1.39
	Pasquet2dr5dnend	2	5	5	3.9	0.99%	1.42
	Pasquet3dr5dnend	3	5	5	1.6	0.98%	1.45
Pasquet3i	Pasquet3i	0	3	3	-0.7	1.03%	1.24
	Pasquet3i0dr2dn	0	2	3	0.8	0.95%	1.24
	Pasquet3i1dr2dn	1	2	3	1.0	0.85%	1.23
	Pasquet3i1dr3dnbeg	1	3	3	-2.6	12.56%	2.98
	Pasquet3i1dr3dnend	1	3	3	-0.7	0.9%	1.26
	Pasquet3i2dr3dn	2	3	3	11.3	2.0%	1.73
	Pasquet3i3dr3dn (1dr after 2$^{\text{nd}}$ i)	3	3	3	16.4	5.25%	1.96
	Pasquet3i0dr4dn	0	4	3	1.8	1.08%	1.26
	Pasquet3i1dr4dnbeg	1	4	3	16.2	3.82%	1.63
	Pasquet3i1dr4dnend	1	4	3	-1.9	0.95%	1.25
	Pasquet3i2dr4dn (1-2/3-4)	2	4	3	3.0	0.89%	1.25
	Pasquet3i2dr4dnend	2	4	3	4.5	1.02%	1.29
Pasquet-simplf	Pasquet-simplf	0	2	3	2.6	1.14%	1.35
	Pasquet-s1dr2dn	1	2	3	1.7	0.94%	1.22
	Pasquet-s2dr2dn (1dr after 2$^{\text{nd}}$ i)	2	2	3	5.0	1.35%	1.31
	Pasquet-s0dr3dn	0	3	3	0.4	1.08%	1.25
	Pasquet-s1dr3dnend	1	3	3	0.5	1.0%	1.25
	Pasquet-s2dr3dn	2	3	3	11.9	2.79%	1.99

Table 7: Results of the tests, grouped by the baseline they belong to and sorted in ascending number of dense layers and number of dropout layers.