Applying Deep Neural Networks (DNN) for Measuring Photometric Redshifts from Galaxy Images: Preliminary Study

M R I Syarifudin¹, M I Hakim¹ and M I Arifyanto¹

¹Department of Astronomy, Institut Teknologi Bandung, Bandung, West Java, Indonesia

E-mail: ilmi.mris@gmail.com

Abstract. In the cosmological and extragalactic study, distance to a galaxy is an important parameter, by knowing the distance, we can find any other physical parameter such as mass, luminosity, star formation rate, and metallicity. By applying the specific cosmological model, we can measure a distance from the redshift. The exact redshift can only be measured by using the spectroscopic technique (Doppler effect), but spectroscopic observation limited to brighter objects and numbers of objects in a single field of view (FoV). While photometric observation can capture fainter objects and more objects in a single FoV. Measurements of photometric redshift could have been done by comparing the SED curves of the elliptical galaxy with known spectroscopic redshifts from other elliptical galaxies which we want to find the photometric redshift. Another method is to do linear or non-linear regression, by assuming the redshift is a function of magnitude in each band-pass filter. Therefore, we propose a technique that using full galaxy images in each measured bands and machine learning method for measuring photometric redshift. We pass entire multi-band galaxy images into the machine learning architecture to get an estimated redshift. In this work, we use galaxies images at 0 ≤ z ≤ 1 from SDSS DR 10 as the datasets and we use DenseNet, one of the Deep Neural Networks (DNN) architecture.

Keywords: Photometric-redshifts, Galaxies, Machine Learning, Deep Neural Networks

1. Introduction
The exact redshift only could measure by using the spectroscopic technique (Doppler effect), but spectroscopic observation limited to brighter objects and numbers of objects in a single field of view (FoV). While photometric observation can capture fainter objects and more objects in a single FoV [1,2]. We used data from telescope survey Sloan Digital Sky Survey (SDSS). SDSS Data Release 10th had 859,322 galaxies spectrums and 208,478,448 galaxies images [3]. Photometric redshift first measured by [4]. He used SED pattern of the elliptical galaxy which known the redshift as a template to measure the direct shifting in the x-axis (wavelength) of another SED from an elliptical galaxy, thus we get the redshift of that galaxy relative to the template. [5] created a new method that inspired by [4], they made SED templates from the galaxy that known redshift each galaxy morphology. [6] used linear and non-linear regression method to estimate the redshift, using assumption redshift was the function of magnitude. We called those methods as template-fitting methods. [7] used Artificial Neural Networks (ANN) for pattern recognition and as an estimator. They used ANN to find a pattern between magnitudes and redshifts. [8] used other machine learning algorithms (MLAs) called k-nearest neighbors (k-NN) and [9] used the random forest. These methods have similar or better
performances when a large spectroscopic training set available. The accuracy of the output photometric redshifts from these methods is limited by the photometric measurements [10].

Nowadays, DNN techniques have revolutionized the field of image recognition. The advantages of DNN is no longer required manual features extraction and feature selection. [11] showed that Deep Convolutional Neural Networks (CNN) were able to provide accurate photometric redshifts from multichannel images, instead of extracted features, taking advantage of all the information contained in the pixels, such as galaxy surface brightness and size, disk inclination, or the presence of color gradients and neighbors. In this paper, we present a DNN model for estimation of photometric redshifts and their associated PDF using the deep learning framework Keras\(^1\) and TensorFlow\(^2\). In contrast to previous studies, our input consists of the following magnitude in \(g, r, i\) band-pass and we use Densely Connected Convolutional Networks (hereafter DenseNet) designed by [12]. The paper is organized as follows. In Section 2, we describe the data used in this study and the pre-processing steps to prepare the images for the DNN. In Section 3, we introduce the CNN concepts and the DenseNet architecture. In Section 4, we present the analysis and the results for the estimation of photometric redshifts and associated PDFs. In Section 5, we conclude and discuss.

2. Methodology

The galaxy data in this study are drawn from the SDSS Data Release 10 [3]. We retrieved 1,918,221 sources classified as galaxy, with the photometric selection criteria following [11]. In detail, we run the MySQL query as shown in the appendix in the CasJobs server. We took ~60000 random sample from the sources before with the spectroscopic redshift below 2 and only got 59861 galaxies downloaded successfully in the following three photometric bands; \(g, r, i\) using python module Astroquery\(^3\). Each pixel in the FITS images file has a resolution of 0.396 arc seconds and contain the measured flux which has been calibrated with flat fielding and sky subtraction. For every entry, we do image registration for the three SDSS filters to resample to a common pixel grid and stack all the available image data using python module Astroalign\(^4\). We rely on the WCS parameters in the input image headers for the astrometry. All pixel fluxes are converted to pixel magnitudes following [13]. We apply a further extinction correction to account for galactic dust using the maps of [14] which is available from the \texttt{photoObjAll} table in the CasJobs server. The extinction corrections are subtracted from magnitude in each pixel in the corresponding FITS image file. We choose to use images of size 72x72 pixels, corresponding to 28.5 arcseconds on a side. The result is 72x72x3 pixel data cube in a gnomonic projection centered on the galaxy coordinates. We convert data cube into RGB images following [15] using python module Astropy. In the term classification problem in supervised learning, the data have a label. We group the value of the spectroscopic redshift from 0.00 to 1.00 into 100 class with bins 0.01 as the labels (e.g. the label from a galaxy with \(z = 0.54123\) is 54). Then, we randomly shuffle and subdivide the sample into training, and test sample by the ratio 70:30. Before we train the MLA on the training sample, we apply k-folds cross-validation with \(k=7\). This approach randomly dividing the set of training sample into \(k\) groups, or folds, of approximately equal size. The first fold is treated as a validation set, and the MLA train on the remaining \(k\) - 1 folds then yield a model. To obtain the predictions of the photometric redshift, we pass the test sample through the model. Fig. 1. show the distribution of the spectroscopic redshift from train and test sample, with these redshift distributions produce a fair estimate of the ability of the MLA to predict redshifts for other galaxies which are representative of the training sample.

---

1 https://keras.io
2 https://www.tensorflow.org
3 https://github.com/astropy/astroquery
4 https://github.com/toros-astro/astroalign
2.1. Deep neural networks

Deep neural networks (hereafter DNN) are based on standard neural networks. DNN depart from these simple neural networks by constructing many hidden layers, with many multiples connected neurons per layers. The power of DNN comes from advances in how the many connections between the many millions of neurons are trained. Previously the many millions of connections would quickly over fit even large training sample, and thereby lose the predictive power [11]. The ways to avoid overfitting model were use data augmentation and add regularization like Dropout and L1/L2 regularization [16].

![Graph showing redshift number distribution of training (blue line) and test (green line) galaxies.](image)

*Fig. 1.* The redshift number distribution of training (blue line) and test (green line) galaxies used in this work. The stepped lines represent the classification bins which are of width 0.01.

2.2. Convolutional neural networks

DNN architecture which can process images as input called Convolutional Neural Networks (hereafter CNN). CNN are typically composed of a number of convolutional and pooling layers followed by fully connected layers. A convolutional layer operates on a data cube, and computes one or several feature maps, also stored as a data cube. For the first convolution layer input data cube is typically a multichannel or multispectral image. Subsequent layers operate of feature maps from previous layers. The pooling layers have function to progressively reduce the spatial size of the representation to reduce the amount of parameters and computation in the network, and hence to also control overfitting. Neurons in fully connected (FC) layer have full connections to all activations in the previous layer, as seen in regular NN. The FC layers are in charge of further processing the features extracted by convolutional layers before, for classification or regression. In this work, we choose to handle the estimation of photometric redshift as a classification problem. The galaxy at the center of the input multichannel image belong to a single class. We applied *softmax* activation function [17] to the output layer, then the sum of the probability must be 1.

2.3. Densely Connected Convolutional Networks (DenseNet)

DenseNet is a network architecture where each layer is directly connected to every other layer in a feed-forward fashion (within each dense block). For each layer, the feature maps of all preceding layers are treated as separate inputs whereas its own feature maps are passed on as inputs to all subsequent layers [12]. We choose DenseNet with hyperparameter as follow; growth rate = 12, depth = 40, number of dense block = 3. We also use regularization the dropout technique with rate 0.4. In total the number of trainable parameter is 1,100,428.
2.4. Photometric redshift estimation
We estimate the photometric redshift of a given galaxy by computing its expected value from its PDF $P(z_k)$:

$$z_{phot} = \sum_{k=1}^{N_c} z_k P(z_k)$$

(1)

where $z_k$ is the midpoint of value of the $k$-th redshift bin. Assuming that $P(z_k)$ is reliable over the whole redshift range.

3. Results and Discussion
We train the MLA with two scenarios. First, we train the MLA only on half of total sample dataset and second we train the MLA on all sample dataset. We want to know how the difference accuracy between the model trained only with around 21 K galaxies and the model trained with around 42 K galaxies. Each scenario, we trained model use 7-folds cross validation and we saved the model every validation accuracy has improvement, we called the early stopping model or checkpoint model and model which saved at the latest iteration of cross-validation we called as CV model.

3.1. Metrics
To assess the quality of the photometric redshift, we adopt the statistics following [11]; $\Delta z = z_{photo} - z_{spec}$ as residual vector, $\mu$, $\sigma_{68}$, $\sigma_{95}$, $\eta$, respectively correspond to the median value of $\Delta z$, the 68% spread of $\Delta z$, the 95% spread of $\Delta z$, and the outlier fraction $|\Delta z|/\sigma_{spec} > 0.15$. The statistics for the performance of the models doing prediction on the testing sample reported in Table 1, we include Hoyle’s work at the first row as comparison. See CV model of 50% sample and CV model of 100% sample from Table 1, we know that the model trained with more sample is better than model trained with less sample. If we look only at the model of 50% sample, checkpoint model performs better than CV model. But, if we look at the model of 100% sample, checkpoint model does not perform better than CV model even if checkpoint model had the best validation accuracy.

| Model                        | N training sample | N testing sample | $\mu$  | $\sigma_{68}$ | $\sigma_{95}$ | $\eta$  |
|------------------------------|-------------------|------------------|--------|---------------|---------------|--------|
| DNN [11]                     | 33,167            | 27,433           | 0.000  | 0.030         | 0.100         | 1.71%  |
| CV model of 50% sample       | 20,952            | 8,979            | 0.008  | 0.102         | 0.204         | 6.72%  |
| Checkpoint model of 50% sample| 20,952            | 8,979            | 0.013  | 0.086         | 0.171         | 3.35%  |
| CV model of 100% sample      | 41,902            | 17,959           | 0.004  | 0.089         | 0.177         | 3.68%  |
| Checkpoint model of 100% sample| 41,902           | 17,959           | 0.013  | 0.089         | 0.179         | 4.14%  |

3.2. Histogram of the estimated photometric redshift
We plot the distribution of spectroscopic redshift from testing sample over the distribution of estimated photometric redshift, see Fig. 2. The shape of the plot distribution of $z_{photo}$ and $z_{spec}$ look similar. However, at the upper, there are missing value at range $z_{photo}$ 0.6 – 1.0, and imply over estimate at range $z_{photo}$ 0.5 – 0.6. That happened because lack of training sample at range 0.6 – 1.0. Meanwhile, at the bottom, the estimated photometric redshifts distributed more likely the spectroscopic redshifts distribution and $z_{photo}$ at range 0.6 – 1.0 are more filled. With this visualization, we could say that model trained with more training sample will performed better than model trained with less training sample. Also, at the bottom, we could see that Checkpoint model of 100% sample more filled and distributed likely the spectroscopic distribution than CV model. So, we also could say that Checkpoint model is better than CV model.
3.3. Plot $z_{\text{photo}}$ against $z_{\text{spec}}$

We use plot $z_{\text{photo}}$ against $z_{\text{spec}}$ to visualize accuracy of the model. More closer data point ($z_{\text{spec}}, z_{\text{photo}}$) to the diagonal line (which mean $z_{\text{photo}} = z_{\text{spec}}$) means more accurate the estimation comes from the model. From Fig. 3. at the upper either at the bottom, we could say that the Checkpoint models have better accuracy than the CV models from the indication of density near at the diagonal line, blue show the low density and red show the high density.

3.4. Histogram of the residual vector

We also use histogram to visualize the distribution of the residual vector. From Fig. 4. we could judge the best model from how high the bar at the median point or at $\Delta z = 0.0$ and also from the spread of $\Delta z$. The model which have the highest point at $\Delta z = 0.0$ is the Checkpoint model of 100% sample. The spread of $\Delta z$ between the Checkpoint model of 50% sample, the CV model of 100% sample, and the Checkpoint model of 100% sample are more likely equal. So, from this visualization we could say the best model is the Checkpoint model of 100% sample.

In this paper we have presented DenseNet, one of the DNN architecture. Because of DenseNet using dense block, we could reduce the numbers of trainable parameter, so it can avoid the risk of overfitting. We used DenseNet with the following hyperparameters; growth rate = 12, depth = 40, number of dense block = 3, and we use the dropout with rate 0.4, thus we only had 1,100,428
parameters. We compiled DenseNet model using Keras and TensorFlow. We had 59,861 sample of the galaxy images and we would to test the DenseNet performance when trained with all of the sample and with half of the sample. We group the value of the spectroscopic redshift from 0.00 to 1.00 into 100 class as the labels. We subdivide the sample into 70% for training sample and 30% for testing sample. We used 7-folds cross-validation, which means we use 6/7 of the training sample for training and 1/7 of the training sample for validating the model. We trained the model using workstation with Intel Xeon E5 8 cores, 8GB of RAM, and GPU Quadro M2000 4GB. We evaluate the model using the testing sample. We pass the testing sample through the trained DenseNet models and thus yield each sample had the PDF. We estimate the photometric redshift from the expected value of PDF. We calculated the metrics to measure the performance of DenseNet models and we visualized the estimated photometric redshift in the following several plots; histogram of the estimated photometric redshift, plot versus, and histogram of residual vector. From the metrics and the visualizations, we concluded more data sample will give the model with better accuracy or better performance and the model saved when had the best validation accuracy is the highly recommended model. Our models do not have much better performance than Hoyle’s model. In the future work, we will try to vary the hyperparameter of DenseNet, add more sample data, and do cross-validation to find the best model.

Acknowledgments
I would like to thank Hakim L. Malasan, Hesti T. R. Wulandari, and Yayan Sugianto from Departement of Astronomy, Institut Teknologi Bandung for helpful conversations and advices, particularly when this work presented for thesis project. Then, we acknowledged Program Penelitian, Pengabdian kepada Masyarakat, dan Inovasi (P3MI) for a funding travel grant to SEAAN 2018.

Appendix A. MySQL data Query
We select data from the SDSS CasJobs website by running the following MySQL query in the Data Release 10:

```
select p.objid, s.specobjid, s.ra, s.dec, s.z as spec_z, s.zerr as err_spec_z, p.dered_u, p.dered_g, p.dered_r, p.dered_i, p.dered_z, p.PETRORAD_R, p.extinction_g, p.extinction_r, p.extinction_i, p.extinction_z into mydb.DR10_DNN from Specobjall s join photoPrimary p on (s.bestobjid = p.objid) and p.devRad_r >0 and p.devRad_r <30 and p.dered_r >0 and p.dered_r < 22 and s.z >0 and s.z<2 and s.zerr >0 and s.zerr<0.1 and p.expRad_r >0 and p.expRad_r <30 and p.type=3
```

References
[1] Smail, I, Hogg, D W, Yan, L, & Cohen, J G 1995 ApJ 449
[2] Newman, J A, Cooper, M C, & Davis, M 2013 ApJS 208 5
[3] Ahn, C P, Alexandre, R, Prieto, et al. 2014 ApJS 211 17
[4] Baum, W A 1962 Problems of extra-galactic research 15 390
[5] Loh, E D, & Spillar, E J 1986 ApJ 303 154
[6] Connolly, A J, Csabai, I, Szalay, A S, et al. 1995 ApJ 110 2655
[7] Collister, A. A., & Lahav, O. (2004). PASP 116 345-351
[8] Csabai, I, Dobos, L, Trencséni, M, et al. 2007 Astronomische Nachrichten 328 852
[9] Carliles, S, Budavári, T, Heinis, S, Priebe, C, & A S 2010 ApJ 712 511
[10] Pasquet J, Bertin E, Treyer M, et al. 2018 A&A 621
[11] Hoyle, B 2016 Astronomy and Computing 16 34-40
[12] Huang, G, Liu, Z, Maaten, L V D, et al. 2017 Densely connected convolutional networks IEEE Conference on Computer Vision and Pattern Recognition (CVPR)
[13] Lupton, R H, Gunn, J E, & Szalay, A S 1999 AJ 118 1406-1410
[14] Schlegel, D J, Finkbeiner, D P, Davis, M, 1998 ApJ 500 525–553
[15] Lupton, R, Blanton, M R, Fekete, G, et al. 2004 PASP 116 133-137
[16] Nielsen, M A, 2015 Neural Networks and Deep Learning Determination Press
[17] Bridle, J S 1990 Neurocomputing, ed. F. F. Soulié & J. Hérald (Berlin, Heidelberg: Springer Berlin Heidelberg) 227–236