Proceedings of the 5th International Conference on Computing and Data Science
DOI: 10.54254/2755-2721/18/20230977

The investigation of transferability related to the ImageNet-based transfer learning model in galaxy classification

Harry Xu

Mission San Jose High School, Fremont, CA 94539, U.S.A.

harxu456@fusdk12.net

Abstract. The phenomenon of transfer learning, specifically the transferability of ImageNet, in the context of galaxy datasets has been relatively under-researched. This study seeks to address this gap by employing the Galaxy10 DECals dataset, which is a 10-classification dataset. Three classic models, namely MobileNet, VGG, and ResNet, were developed and customized to the current dataset by modifying the number of neurons in the last layer. The experimental phase is divided into three parts, including the comparison between the use of ImageNet and non-ImageNet, model performance comparison, and confusion matrix analysis. The results demonstrate that the utilization of ImageNet produces better outcomes, with the MobileNet model exhibiting the highest performance. Further analysis revealed that the inclusion of ImageNet weights can enhance the classification accuracy of some data. Although the present study was successful in achieving its objectives, future research should focus on exploring and elucidating the underlying mechanisms driving these findings.

Keywords: ImageNet, transfer learning, convolutional neural network.

1. Introduction

Galaxies are fascinating objects that have captivated astronomers and astrophysicists for centuries. They come in various shapes, sizes, and colours, and understanding their properties is crucial to people's understanding of the universe. Over the years, astronomers have classified galaxies into different types based on their morphology, colours, and other physical properties. However, this process is time-consuming and can be prone to human biases and errors. To solve this issue, finding a more effective approach that is conductive to the galaxy classification prediction is of great necessity.

Historically, the classification of galaxies was based on traditional image processing algorithms that relied on manual identification of morphological features. One such example is the Hubble sequence, proposed by Edwin Hubble in the 1920s [1]. This classification system uses the appearance of galaxy spirals and ellipticals to group them into different categories. This system is still in use today, albeit with revisions and expansions over the years. Another widely used classification is the de Vaucouleurs system, proposed by Gérard de Vaucouleurs in the 1950s [2]. This classification system utilized a more detailed set of criteria to classify galaxies based on their bulge-to-disk ratio, arm structure, and other features. In addition, Holwerda et al. also proposed a CAS system that uses a combination of visual features (such as bulge-to-disk ratio and asymmetry) and machine learning algorithms to classify galaxies into different categories [3].

These algorithms and systems have been widely used in the field of astronomy for many years. However, these traditional image processing methods had limitations such as low accuracy and slow analysis speed. With the recent advancements in deep learning and machine learning, new approaches have been proposed for galaxy classification that can offer higher accuracy and more efficient analysis.

In recent years, with the rapid development of deep learning, Convolutional Neural Networks (CNNs) have become increasingly popular for galaxy classification. Some researches proposed the use of CNNs for galaxy classification, and several studies have demonstrated the effectiveness of this approach. For instance, Zhu et al. employed a deep convolutional neural network to classify galaxy morphologies [4]. They achieved classification accuracies of up to 95%, demonstrating the effectiveness of CNNs for this task. Similarly, M. Huertas-Company also utilized a CNN to classify galaxies based on their visual appearance in the Galaxy Zoo dataset [5]. They achieved classification accuracies of up to 92%, outperforming traditional machine learning algorithms. Zhang et al. also proposed a CNN to classify galaxy morphologies in the Galaxy10 dataset. They achieved classification accuracies of up to 95% [6].

In addition, transfer learning has also been explored for galaxy classification. Naik et al. used transfer learning with pre-trained Imagenet weights to classify galaxy morphologies in the Galaxy10 dataset, achieving classification accuracies of up to 94% [7]. However, most transfer learning approaches are based on the Imagenet dataset, which does not include galaxy-specific features. Therefore, the feasibility of using Imagenet weights for galaxy classification is still an open research question.

This paper aims to answer this question by proposing a novel approach for galaxy classification using deep learning techniques and the Galaxy10 Dark Energy Camera Legacy Survey (DECals) dataset from AstroNN. The Galaxy10 dataset is a large-scale dataset of 17,736 galaxies with 10 distinct morphological classes. This study firstly explores the effectiveness of different deep learning models for galaxy classification. Furthermore, the impact of using transfer learning with different pre-training weights on the performance of the models was also investigated.

2. Method

2.1. Dataset description and pre-processing

The Galaxy10 DECals dataset is a collection of 17,736 coloured images of galaxies. The dataset has columns images with shape (17736, 256, 256, 3), ans, ra, dec, redshift and pxscale in unit of arcsecond per pixel. It covers three different wavelength bands: g, r, and z. The galaxies in the dataset are categorized into ten different classes based on their morphology, with each class containing approximately the equal number of images.

To pre-process the Galaxy10 dataset, several steps were taken before passing it into the following deep learning model for training. First, the images were labelled and resized to a uniform size of 256 × 256 pixels with 3 channels of colour image. Secondly, the labels were converted to categorical 10 classes and desirable float 32 type. Subsequently, they were normalized by dividing each pixel value by the maximum pixel value (i.e. 255) to scale the values to the range [0, 1]. Finally, the labels were transformed into one-hot encoding and the dataset was partitioned into training and testing sets with 70/30 ratio. Some example images of the collected galaxy dataset can be found in Figure 1.

DOI: 10.54254/2755-2721/18/20230977

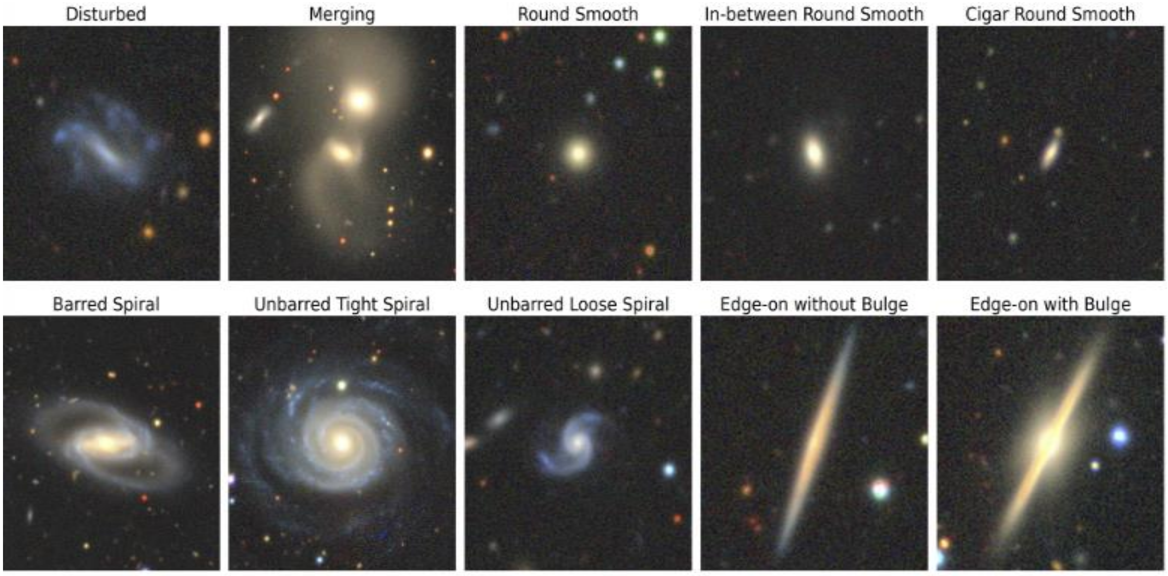

Figure 1. Example images of each class from Galaxy10 DECals.

2.2. Convolutional neural network models

CNNs are a popular deep learning model frequently employed for image classification, object detection, and other computer vision tasks. The fundamental concept underlying CNNs involves the acquisition of hierarchical representations of image features, through a sequence of convolutional, pooling, and fully connected layers. In particular, the convolutional layer is the key element of a CNN, responsible for discerning local features from the input image with the aid of a set of adaptable filters. These filters slide over the input image, executing a dot product with a region of pixels to produce a feature map. The convolutional layer can learn multiple features, such as edges, corners, and textures, through the utilization of various filters. The pooling layer serves to downsample the feature maps and lower the spatial dimensionality of the data while maintaining significant features. Max pooling, the most common type of pooling, selects the highest value within a small region of the feature map and discards the remaining values. This operation minimizes the dimensionality of the feature maps, thereby simplifying subsequent layer processing. The fully connected layer classifies the input image based on the extracted features. It receives the flattened feature map output from the convolutional and pooling layers and passes it through a series of fully connected layers, which function as a conventional neural network classifier. These layers learn to map the features to the output classes, utilizing techniques like dropout and activation functions to avoid overfitting.

MobileNet is a deep neural network architecture specifically designed for efficient computation on mobile and embedded devices. MobileNet was created by Howard et al. in 2017 [8]. The fundamental concept underlying MobileNet is to leverage depthwise separable convolutions to decrease the computational complexity of the network, while maintaining a high degree of precision. Depthwise separable convolutions bifurcate the standard convolution operation into two independent layers: a depthwise convolution that filters each input channel individually, and a pointwise convolution that amalgamates the results of the depthwise convolution across channels. This technique reduces the number of parameters and computations required by the network, resulting in enhanced efficiency and speed of execution. The MobileNet architecture is a potent and efficient deep neural network that is particularly suitable for mobile and embedded devices. Its computational efficiency and high accuracy render it a popular choice for a wide range of computer vision tasks, including object detection, image recognition, and more.

ResNet-50 was introduced by Kaiming He, et al. [9]. The ResNet architecture was designed to tackle the vanishing gradient problem commonly encountered in training very deep neural networks. ResNet-50 consists of 50 layers and introduces skip connections to enable training of very deep networks. The

DOI: 10.54254/2755-2721/18/20230977

skip connections allow gradients to flow directly through the network, bypassing the intermediate layers. This helps to mitigate the vanishing gradient problem and enables the training of very deep networks. ResNet-50 uses residual blocks, which consist of two or three convolutional layers and a skip connection, to enable the efficient training of very deep networks. It has also been used as a starting point for transfer learning in many computer vision applications. As of 2023, ResNet-50 remains widely used in computer vision research and applications and is considered a standard benchmark architecture for various image recognition tasks. Numerous variants of ResNet, including ResNet-101 and ResNet-152, have also been developed since the introduction of ResNet-50.

The Visual Geometry Group (VGG) introduced the VGG16 convolutional neural network architecture [10]. The VGG16 architecture comprises 16 layers, including 13 convolutional layers and 3 fully connected layers. It is renowned for its utilization of 3×3 convolutional filters throughout the network, enabling it to learn intricate features at different scales. The VGG16 architecture has delivered exceptional performance in several image recognition tasks, such as the ImageNet Large Scale Visual Recognition Challenge. The VGG16 architecture was developed to investigate the influence of network depth and convolutional filter size on image recognition performance. The researchers discovered that increasing network depth and using smaller 3×3 filters was more advantageous than using larger filters. Currently, the VGG16 architecture remains extensively employed in computer vision research and applications, and has been deployed in various tasks, including object detection, semantic segmentation, and image captioning.

2.3. Implementation details

The TensorFlow library was imported, and this study loaded the CNN model (i.e. MobileNet, ResNet-50 and VGG16) with ‘weights’ parameter set to “imagenet” to use the pre-trained weights from the ImageNet dataset shown in Figure 2. The ‘include_top=False’ parameter specifies that the final classification layer of the pretrained model would not be considered, as the number of neurons in the final layer of the neural network needed to be adapted to the current task. The input_shape parameter specifies the size of the input images, which is set to (224, 224, 3) for CNN models. Next, the study details the process of adding new layers to the CNN models by creating a new TensorFlow model and adding CNN model (i.e. MobileNet, ResNet-50 and VGG16) as a layer. Then a Flatten layer was included to flatten the output from the CNN models, followed by one fully connected layer with 10 units, respectively. The final layer uses a softmax activation function to output the class probabilities. Moreover, the weights can be adjusted to either use pre-trained weights from the ImageNet dataset or train the model from scratch. In the case of pre-trained weights, the CNN layers' trainable parameter is set to False, allowing for transfer learning with the pre-trained weights. Conversely, to train the model from scratch, the trainable parameter remains True. In the compilation and train the model, the ‘x_train’ and ‘y_train’ are the training data and labels, respectively, and ‘x_val’ and ‘y_val’ are the validation data and labels, respectively.

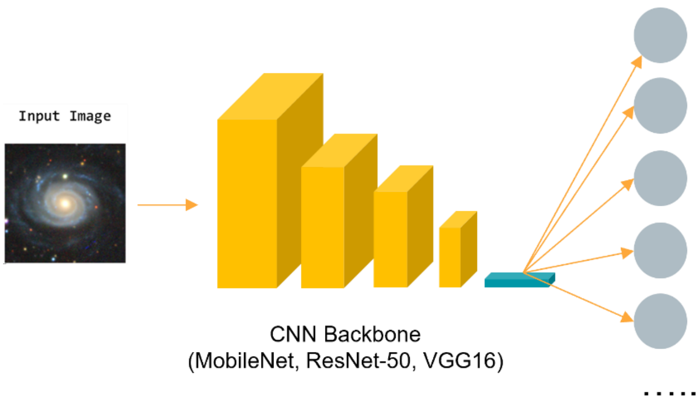

Figure 2. The structure of the CNN model used in this study.

DOI: 10.54254/2755-2721/18/20230977

3. Result and discussion

3.1. The performance of different CNN models based on various configuration

Based on the findings presented in Figure 3 and Table 1, it can be inferred that the MobileNet model trained on Imagenet dataset exhibits the highest level of performance with an accuracy value of 0.8226. In comparison, the MobileNet model without Imagenet weighs demonstrates a relatively lower performance with an accuracy value of 0.6861, signifying a performance improvement of 0.1365 or 19.8% with the use of Imagenet weights. Similar conclusions can be drawn for VGG and ResNet models as well. Overall, the utilization of Imagenet weights appears to enhance the performance of the deep learning models.

Upon careful examination of the loss and accuracy curves during the training phase, as depicted in Figure 4 through Figure 15, it is evident that deep learning models utilizing Imagenet weights exhibit a comparatively lower overall loss on both the training and validation datasets, accompanied by higher accuracy when compared to models without Imagenet weights. Furthermore, models incorporating Imagenet weights tend to demonstrate a higher tendency towards overfitting, as opposed to those without Imagenet weights.

The explanation for the potential superior performance of Imagenet compared to models without Imagenet weights was provided. This article speculates that although there are no galaxy images in the Imagenet dataset, there are some similar features that can be transferred to this galaxy classification task. For instance, Imagenet contains circular shape features and related color features, which are similar to the overall features of objects in galaxies. Therefore, the combination of these features could lead to an improvement in the classification of galaxy images. Another possible reason is that the weights (i.e. w) and biases (i.e. b) of models without Imagenet weights may start from a unfavourable starting point at the beginning when training the model, and may require more time to converge to the global or local optimal values during optimization. However, models with Imagenet weights have w and b that have been trained, so they may have a better starting point that is closer to the global or local optimal values. Therefore, the models can converge to the optimal values more quickly, potentially leading to improved performance in the classification task.

Based on the results presented in Table 1 and Figure 3, it is evident that the MobileNet model exhibits superior performance in comparison to ResNet, while ResNet outperforms VGG. Notably, the ImageNet accuracy of MobileNet (0.823) surpasses that of ResNet (0.625) and VGG (0.142), with a difference of 0.198 between MobileNet and ResNet. Regarding ResNet and VGG, the difference amounts to 0.483. Similar trends can be observed in the precision and F1-score metrics.

The article posits that the observed performance disparities among the models can be primarily attributed to their varying numbers of trainable parameters. Specifically, MobileNet represents a lightweight model with the smallest parameter count, thereby enabling its parameters to be fully trained with a relatively small number of training iterations. Conversely, models such as ResNet and VGG feature an excessive number of parameters, impeding convergence during training or necessitating prolonged training periods to attain optimal performance levels.

Table 1. The performance of different models.

Model Name	ImageNet	Accuracy	Precision	F1-Score
VGG16	ON	0.142	0.136	0.142
ResNet50	ON	0.625	0.703	0.632
MobileNet	ON	0.823	0.849	0.823
VGG16	OFF	0.157	0.025	0.043
ResNet50	OFF	0.564	0.698	0.575
MobileNet	OFF	0.686	0.679	9.663

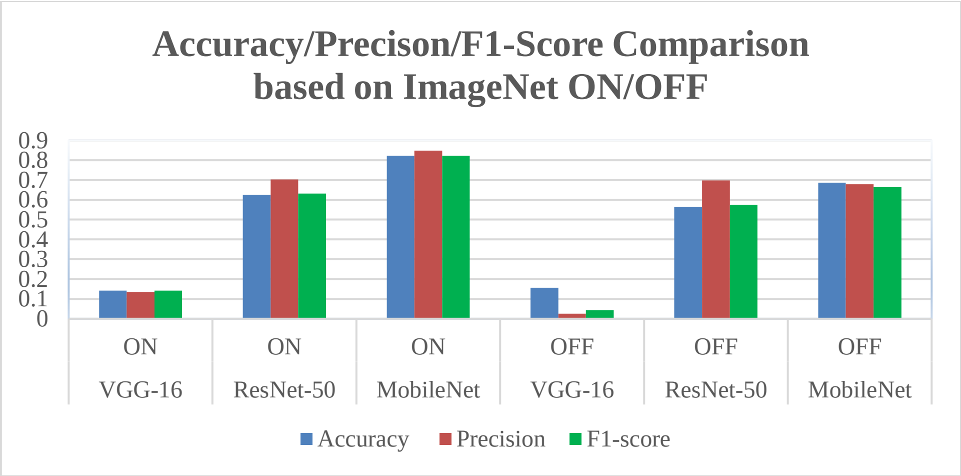

Figure 3. Accuracy/Precision/F1-Score comparison based on ImageNet ON/OFF.

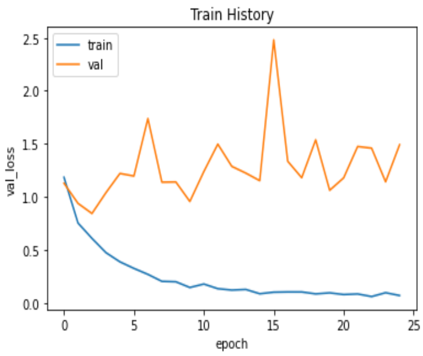

Figure 4. Train and validation loss (MobileNet with Imagenet weight).

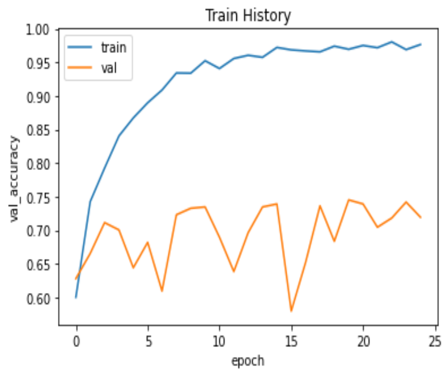

Figure 5. Train and validation accuracy (MobileNet with Imagenet weight).

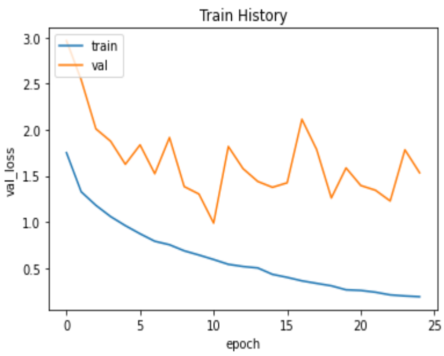

Figure 6. Train and validation loss (MobileNet without Imagenet weight).

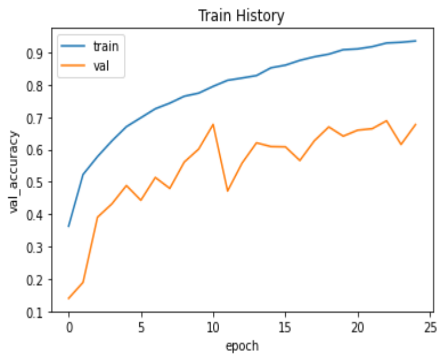

Figure 7. Train and validation accuracy (MobileNet without Imagenet weight).

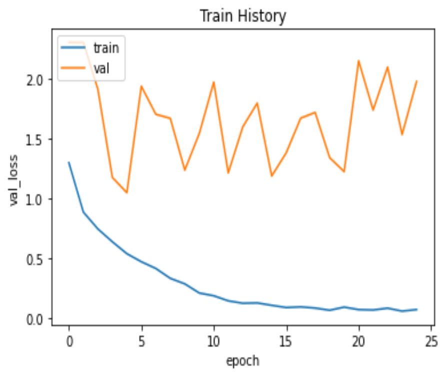

Figure 8. Train and validation loss (ResNet with Imagenet weight).

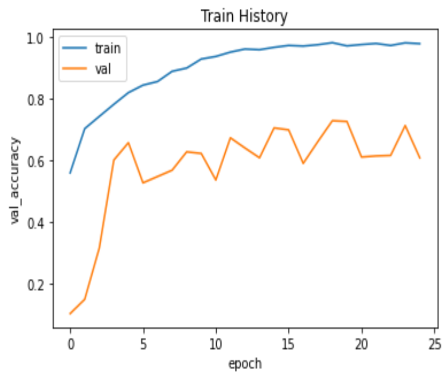

Figure 9. Train and validation accuracy (ResNet with Imagenet weight).

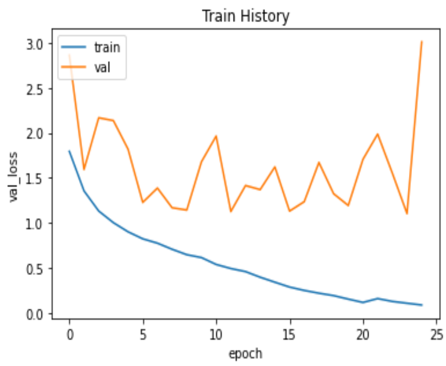

Figure 10. Train and validation loss (ResNet without Imagenet weight).

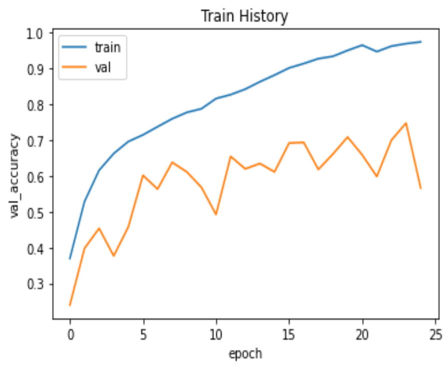

Figure 11. Train and validation accuracy (ResNet without Imagenet weight).

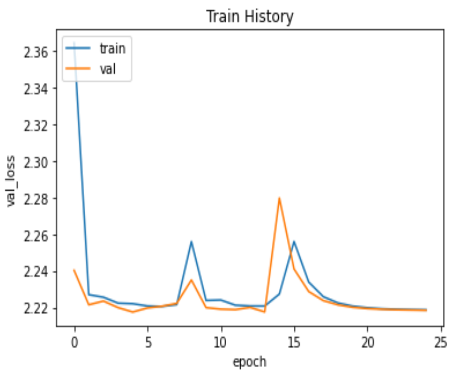

Figure 12. Train and validation loss (VGG with Imagenet weight).

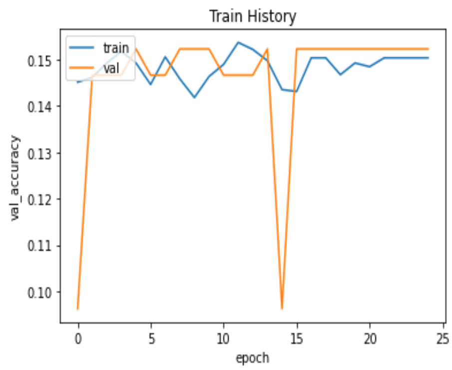

Figure 13. Train and validation accuracy (VGG with Imagenet weight).

DOI: 10.54254/2755-2721/18/20230977

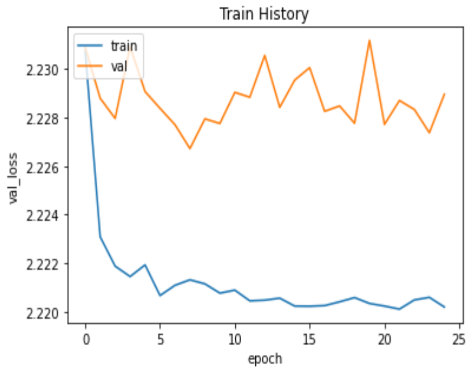

Figure 14. Train and validation loss (VGG without Imagenet weight).

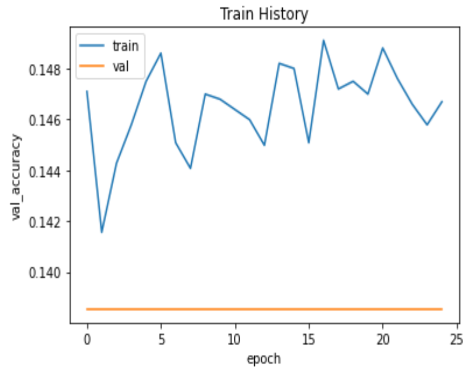

Figure 15. Train and validation accuracy (VGG without Imagenet weight).

3.2. Confusion matrix based on different models

The Figure16 - Figure 21 present 6 confusion matrices corresponding to 3 models based on different weights (i.e. ImageNet, No ImageNet). From the confusion matrix, it can be observed that which categories the model fails to predict. For MobileNet, the model with ImageNet weights is less likely to confuse class 8 and class 7 compared to the MobileNet model without ImageNet weights during classification. This could be due to the fact that the ImageNet weights contain many spiral-related images and features that can be transferred to the current task, making it easier for the MobileNet model based on ImageNet weights to differentiate in this aspect. The ResNet model with ImageNet weights is less likely to predict all categories incorrectly as class 2 compared to the ResNet model without ImageNet weights, possibly because the weight based on imagenet dataset contains the feature of multiple circles, so it is more likely to achieve the better performance. As for VGG, regardless of ImageNet or no ImageNet, the model performs poorly and cannot be trained properly. All categories are predicted as class 8 incorrectly, making it incomparable.

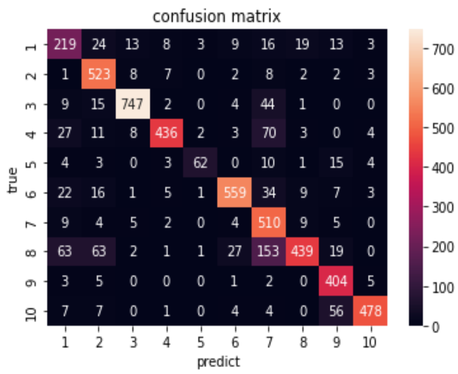

Figure 16. Confusion matrix (MobileNet with Imagenet weight).

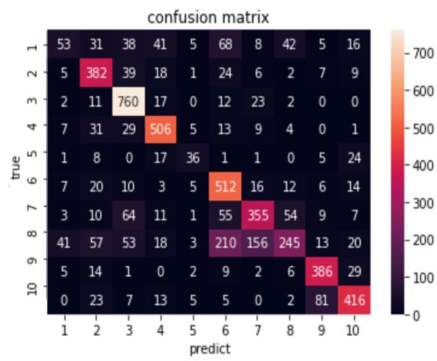

Figure 17. Confusion matrix (MobileNet without Imagenet weight).

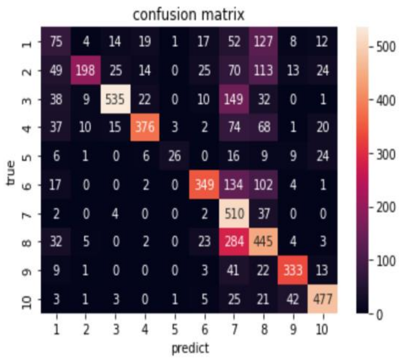

Figure 18. Confusion matrix (ResNet-50 with Imagenet weight).

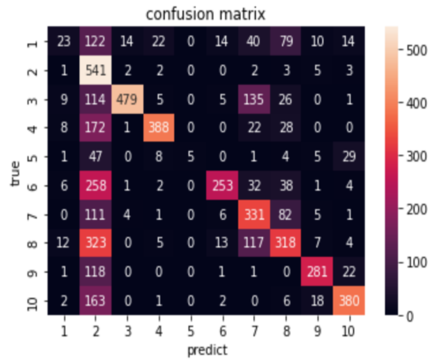

Figure 19. Confusion matrix (ResNet-50 without Imagenet weight).

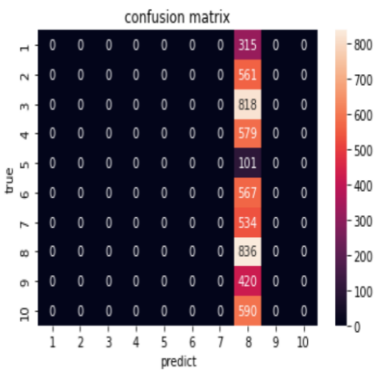

Figure 20. Confusion matrix (VGG-16 with Imagenet weight).

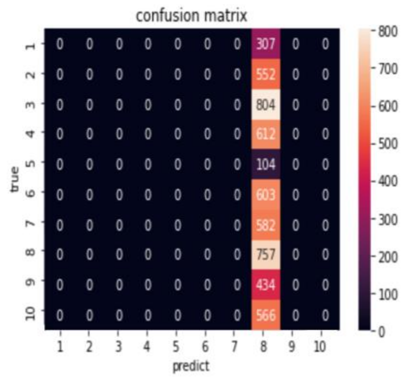

Figure 21. Confusion matrix (VGG-16 without Imagenet weight).

4. Conclusion

This article provides a detailed discussion and analysis of the transferability of ImageNet and no ImageNet weights in the context of a galaxy dataset. To achieve this objective, several classical models, namely VGG, MobileNet, and ResNet, were employed. Two versions of each model were compared, one with ImageNet weights and one without. The experimental findings reveal that ImageNet weights outperform no ImageNet weights, with MobileNet exhibiting the highest performance among the three models. Nevertheless, the study is not without its limitations, as there is a lack of experimental evidence to explicate or corroborate the specific reasons underlying these outcomes. Hence, further research is warranted to address these limitations and shed light on the mechanisms driving these findings.

References

[1] Hubble E P 1926 Extra-galactic nebulae Astrophysical Journal 64: pp. 321-369 Bibcode: 1926 ApJ.64..321H
[2] Gérard d V 1959 Introduction Classification Morphology Qualitative morphology Quantitative morphology Handbuch der physic Volume 53 p275
[3] Holwerda B W et al 2011 Quantified H i morphology–III. Merger visibility times from H i in galaxy simulations Monthly Notices of the Royal Astronomical Society 416(4): pp 2426-2436
[4] Zhu X P et al 2019 Galaxy morphology classification with deep convolutional neural networks Astrophysics and Space Science 364 pp 1-15
[5] Marc Huertas-Company 2018 Automated Morphological Classification of Galaxies with Deep Learning The Astrophysical Journal

DOI: 10.54254/2755-2721/18/20230977

[6] Zhang R et al 2019 Galaxy morphologies in the era of deep learning: a case study with Galaxy Zoo 10 and Illustris simulations Monthly Notices of the Royal Astronomical Society vol 485 no 3 pp 2895–2907
[7] Naik T R et al 2019 Deep transfer learning for star/galaxy classification with the Galaxy10 dataset Monthly Notices of the Royal Astronomical Society vol 484 no 1 pp 282–292
[8] Howard A G et al 2017 Mobilenets: Efficient convolutional neural networks for mobile vision applications arXiv preprint arXiv:1704.04861 2017
[9] He K Zhang X Ren S et al. 2016 Deep residual learning for image recognition Proceedings of the IEEE conference on computer vision and pattern recognition pp 770-778
[10] Simonyan K Zisserman A 2014 Very deep convolutional networks for large-scale image recognition arXiv preprint arXiv:1409.1556