**MORPHOLOGICAL**

**CLASSIFICATION**

**OF**

**GALAXIES**

**Abstract**

This project report represents the morphological classification of galaxies as well as the comparative study of the machine learning Models. A comparative study is performed on transformed data e.g., cropping, downscaling, and augmenting. However, the main purpose of data transformation was to avoid overfitting. Both Grayscale and RGB images are used to achieve high performance. Models that are used for the classification purpose are **Multilayer Perceptron Model** and **Resnet50 Model**. Besides, this problem has also been hosted on Kaggle*[[1]](#footnote-1)* in 2017, for which the highest score ever obtained was ~0.079 (minimum loss). Therefore, my primary goal was to achieve a loss which is less than 0.06 (Root Mean Squared Loss). For that, multiple variations of a model are used and compared. All of these variations are discussed thoroughly in this report.

**Table of contents**

1. **Introduction**
2. Analysis
   1. Galaxy Zoo Project
   2. Dataset
3. Design
   1. Multilayer Perceptron Architecture
   2. ResNet50 Architecture
4. **Implementation**
   1. Data Preprocessing
      1. Crop
      2. Resize
      3. Rescale
      4. Data augmentation
         1. Horizontal Flip
         2. Vertical Flip
         3. Random Rotation
         4. Zoom
         5. Brightness
   2. Model Compilation
   3. Model Fitting/Model Training
      1. MLP Training Analysis
         1. Without Data Augmentation
         2. With Data Augmentation
      2. Resnet Training Analysis
         1. Without Data Augmentation
         2. With Data Augmentation
5. Testing
6. Conclusion and Future Work
   1. Conclusion
   2. Future Scope
7. Bibliography/References

# ****Introduction****

The Universe has always intrigued us by its cosmic beauty and galactic evolution. So, to unveil the actual evolution process along with the mystery of the Universe astrophysicists and astronomers are trying hard. For that, there are many surveys that are incessantly trying to do so e.g., *Dark Energy Survey* ***(DES)[[2]](#footnote-2)*** and *Sloan Digital Sky Survey* ***(SDSS)[[3]](#footnote-3)***. Scientists are perpetually discovering new planets, extra solar planets, galaxies, and other celestial bodies every day with the help of giant telescopes. The world is witness to the great discoveries in astronomy, and one of them is *"The first-ever Image of Black Hole*" **(2017)[[4]](#footnote-4)** by Katie Bouman.

Among all these researches, galaxy morphology is the one that is oldest and significant. The reason is that galaxy morphology reveals evolution history and stellar properties of itself. Galaxies can easily be classified into two main types, namely early-type galaxies (ETGs) and late-type galaxies (LTGs). ETGs include mainly massive elliptical galaxies, and lenticular galaxies without spiral structure and with older populations. LTGs are comprised of spiral galaxies and irregular galaxies with a younger population.

But more and more survey projects e.g., Sloan Digital Sky Survey (SDSS), the Dark Energy Survey (DES) [1]which has images of more than hundreds of millions of galaxies, and the *Large Synoptic Survey Telescope* (LSST)[[5]](#footnote-5)had resulted in a data explosion. Hence, it became unfeasible for experts to classify all these images manually.

Therefore, the researchers came up with the idea of releasing crowdsourcing astronomy projects, where citizen scientists volunteer. The series of Galaxy Zoo projects [2] [3] are one of the successful tools for large-scale morphology analysis. The goal of GZ1 [3] was to classify galaxies into Ellipticals and Spirals. After the success of the GZ1 project, GZ2 [2] came up with detailed features of galaxies than just fundamental, e.g., strong bars, central bulges, edge-on disks, and arm curvature.

Galaxy Zoo 2 [2] comprises detailed morphologies for more than 300 thousand SDSS galaxies (largest and brightest). A more detailed description of the SDSS data and GZ2 catalogue is provided further in the section **2.2**. However, these are small-scale features. Moreover, it is not possible to extract all independent features of galaxies on which the Classification relies. Indeed, a successful approach to this problem is to apply machine learning and computer vision techniques.

 Two models have applied to this data; ***Multilayer Perceptron*** and ***Resnet50***. Both the models have also trained separately on the image data with or without data augmentation. This report describes the whole process and outcomes.

1. **Analysis**

* 1. **Galaxy Zoo Project**

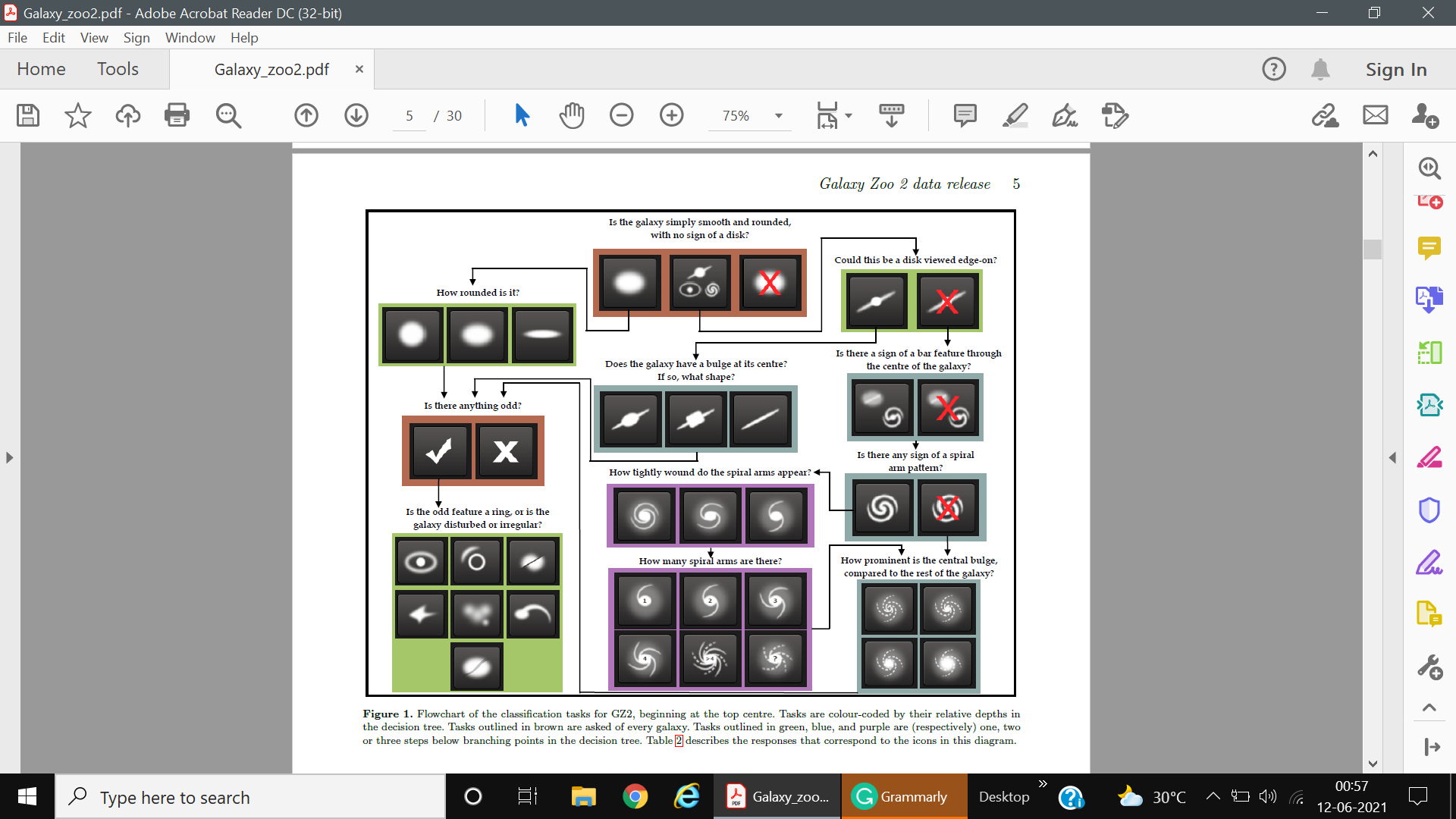
This project was one of the most successful crowdsourcing projects in astronomy. The morphological classification of galaxies became impractical due to the data explosion ensued by some major surveys. Because these major surveys generate data in Terabytes, e.g., DES DR2 [2] includes more than 600 million distinct astronomical objects. These objects are detected in more than 10K co-added image tiles where co-added images are produced from 76,217 single-epoch images with median image-stacking.

Moreover, DES covers southern Galactic cap (about 5000 deg2) in five broad photometric bands, **“grizY**”.

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Question** | **Response** | **Next** |
| 02 | Could this be a disk viewed edge-on? | Yes  No | 09  03 |
| 03 | Is there a sign of a bar feature through the centre of the galaxy? | Yes  No | 04  04 |
| 04 | Is there any sign of a spiral arm pattern? | Yes  No | 10  05 |
| 05 | How prominent is the central bulge, compared with the rest of the galaxy? | no bulge  just noticeable  obvious  dominant | 06  06  06  06 |
| 06 | Is there anything odd? | yes  no | 08  end |
| 07 | How rounded is it? | completely round  in between  cigar-shaped | 06  06  06 |
| 08 | Is the odd feature a ring, or is the galaxy disturbed or irregular? | ring  lens or arc  disturbed  irregular  other  merger  dust lane | end  end  end  end  end  end |
| 09 | Does the galaxy have a bulge at its centre? If so, what shape? | rounded  boxy | 06  06 |
| 10 | How tightly wound do the Spiral arms appear? | tight  medium  loose | 11  11  11 |
| 11 | How many spiral arms are there? | 1  2  3  4  more than four 05  can't tell 05 | 05  05  05  05  05  05 |

**Table1.** This is the decision tree, representing all the questions (11) asked from volunteers. Here, ‘Questions’ are actual questions that were asked consecutively (only if previous question is answered), and ‘Response’ (total 37) shows all the possible responses that can be given by volunteer for corresponding question. ‘Next’ shows the total questions that are supposed to be asked for complete classification.

Therefore, astronomers had commenced this project to speed up the tedious manual classification. In this project, 11 questions were asked to each volunteer where each task had some possible responses, and the volunteer had to choose one among those. Remember that, the term "questions" is referred to as "tasks" in the original GZ2 [2] paper, hence both terms are used interchangeably in this report. Consequently, the ***decision tree***in figure1represents the questions asked from volunteers to find features in galaxies.

****

**Figure1.** This is the flowchart of the decision tree, represented in Table1.

* 1. **Dataset**

The dataset is collected from Galaxy Zoo 2 [2]. It was a citizen science project for large-scale morphological classification with more than 16 million objects, classified. The classification is performed with 304,122 galaxies drawn from the SDSS [4]. The images are gathered from both DR7 Legacy survey having apparent magnitude (m > 17), and SDSS Stripe 82 [4]. All images are *gri* color composite images, each of size (424 × 424) pixels.

The final Dataset is separated into two different sets. One contains 61578 galaxies for training purposes, and the other contains 79975 galaxies for testing purposes. All galaxies are JPG images with pixels described in the above section. Furthermore, there are 37 labels assigned to each galaxy where a label represents the *joint probability* of the particular response to the corresponding task[[6]](#footnote-6)**.**



**Figure2:** 424 × 424 galaxy images

Since the tasks represented in the above decision tree are dependent on the set of some previous tasks, except the first one, hence the corresponding probabilities are not independent either. For e.g., let's say task2 (*"Could this be a disk viewed edge-on?")*with marginal probability p1 is dependent on task1 ("Is the galaxy simply smooth and rounded, with no sign of features or a disk?") with marginal probability p2, then the joint probability of task2 would be p1×p2. The same method has been applied to all the consecutive tasks. Remember that, before this computation, the sum of all likelihoods of each task was equal to 1.

More importantly, all galaxies are located in the *center* of images which came out as an advantage up to an extent. Additionally, most of the images were invariant to rotation, translation, and scaling. These invariances were exploited successfully for data augmentation**.**

Therefore, this seeming to be a classification problem is a regression problem in reality. The reason is that labels are not independent as they are dependent on each other. The dependencies among all 11 tasks are shown below;

|  |  |
| --- | --- |
| **Task** | **Previous Task** |
| **01** | - |
| **02** | 01 |
| **03** | 01,02 |
| **04** | 01,02 |
| **05** | 01,02 |
| **06** | - |
| **07** | 01 |
| **08** | 06 |
| **09** | 01,02 |
| **10** | 01,02,04 |
| **11** | 01,02,04 |

**Table2.** It shows the dependencies among tasks.

1. **Design**

## Multilayer Perceptron Architecture

MLP is an artificial feed-forward neural network with one or more hidden layers. This model takes advantage of chain rule (back propagation) and learns the pattern within the data. The architecture of this model is shown below;

**Input Layer**

Neurons = Image shape

16384

or,

49512

**Hidden layer 1**

4096 **Neurons**

Activation

**‘Relu’**

**Hidden layer 2**

512

Neurons

Activation

**‘Relu’**

**Hidden layer 3 with dropout rate 0.5**

256

Neurons

Activation

**‘Relu’**

**Hidden layer 4**

37

Neurons

Activation

**‘Relu’**

**Custom Output Layer**

37

Neurons

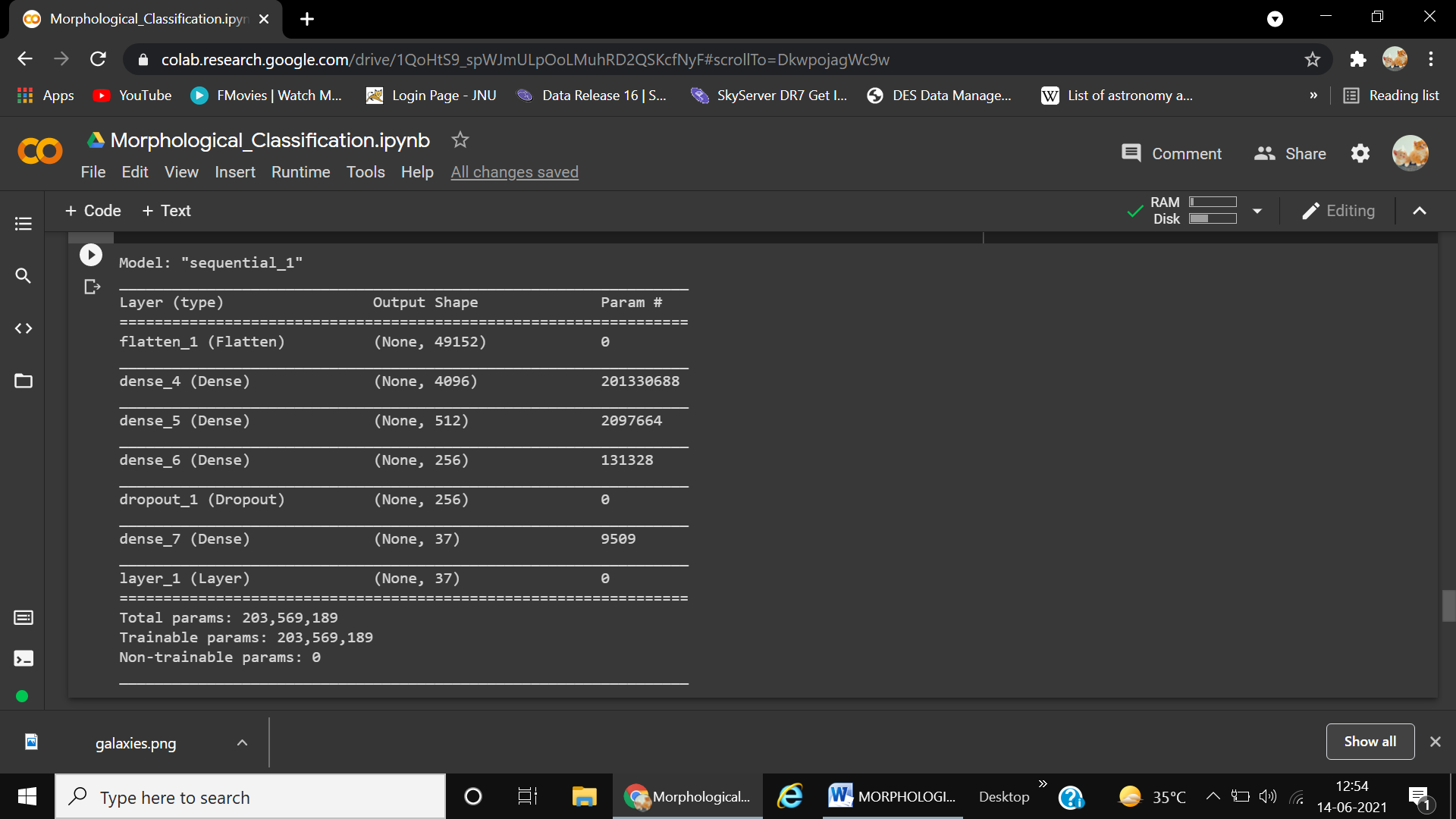
**Figure3:** Multilayer Perceptron architecture

1. **Input Layer/Flatten Layer:**  It is the very first layer of MLP that takes input. It flattens (converting into a 1D array) the 2D or 3D array (array of pixels). So the number of neurons in this layer depends on the shape of the image e.g., a Grayscale image contains a single channel, whereas an RGB image contains three channels. Accordingly, the total neurons will be:

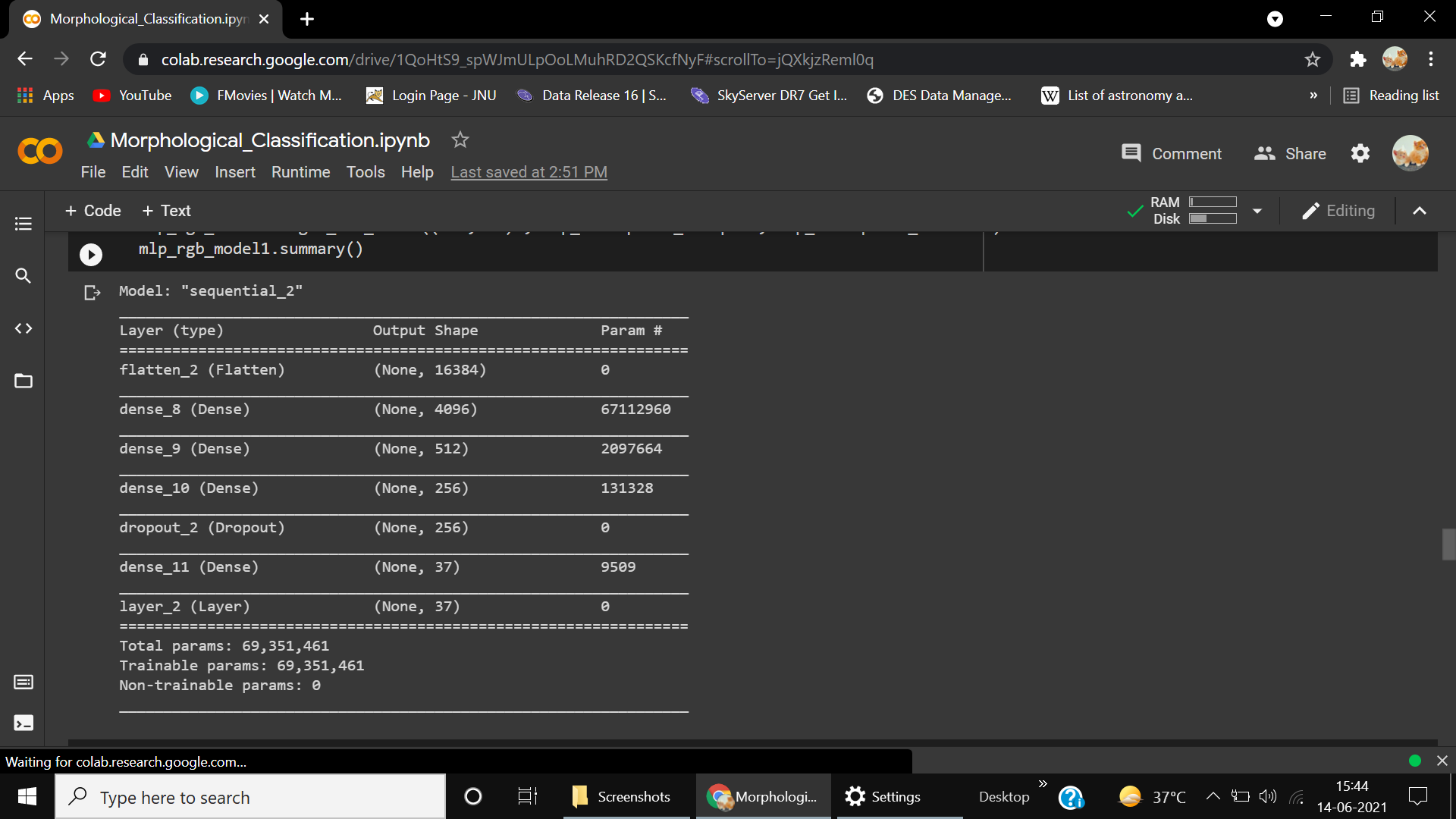
* 4096 (Grayscale images of shape (64, 64))
* 12288 (RGB images of shape (64, 64, 3))

1. **Hidden Layer 1:**It is a fully connected dense layer containing 4096 neurons and a 'Relu' activation function.
2. **Hidden layer 2:**A fully connected dense layer containing 512 neurons with 'Relu' activation function.
3. **Hidden Layer 3:**A dense layer having 256 neurons with 'Relu' activation. In addition, this layer also has a Dropout rate of 0.5 to reduce overfitting.

1. **Hidden layer 4:**This layer is technically an output layer. But, for the different probability distribution in tasks the output of this layer has to be modified. However, it contains 37 neurons with 'Relu' activation. Note that we can’t apply “*Softmax”* or *“Sigmoid”* here as the distribution is not the same for each label.
2. **Output layer:**The final layer of our model contains 37 neurons with explicit customization. Analytically, weights do not need to be initialized with this layer because the performance does seem to increase at all. Hence, I have kept this layer without weight initialization. So basically, it is not a layer, indeed it is an additional 'custom activation function' applied to the previous layer.

******

**Figure4.** MLP summary for RGB Images

****

**Figure5.** MLP model summary for grayscale images

## ResNet50 Architecture

ResNet [5] Model is one of the successful in deep learning history and computer vision. It is an abbreviation of a residual network which is the special kind of neural network. Their specialty is the use of residual blocks or skip-connection blocks. These blocks pass the information through blocks by skipping layers in between two consecutive blocks. This model has been a winner of 'Imagenet' 2015 [6] - the reason for its invention.

ResNet is a Convolutional neural network [7] with deep layers consisting of the blocks of shallow CNNs. Training a very deep layer network was hard back then due to the infamous vanishing gradient problem.

**ResNet-50 without top layers**

**Pooling:** Average

**Last layer:** 2048 neurons with dropout rate 0.5

**Weights:** Imagenet

**Hidden layer 4**

37

Neurons

Activation

**‘Relu’**

**Custom Output Layer**

37

Neurons

&

**Custom** activation

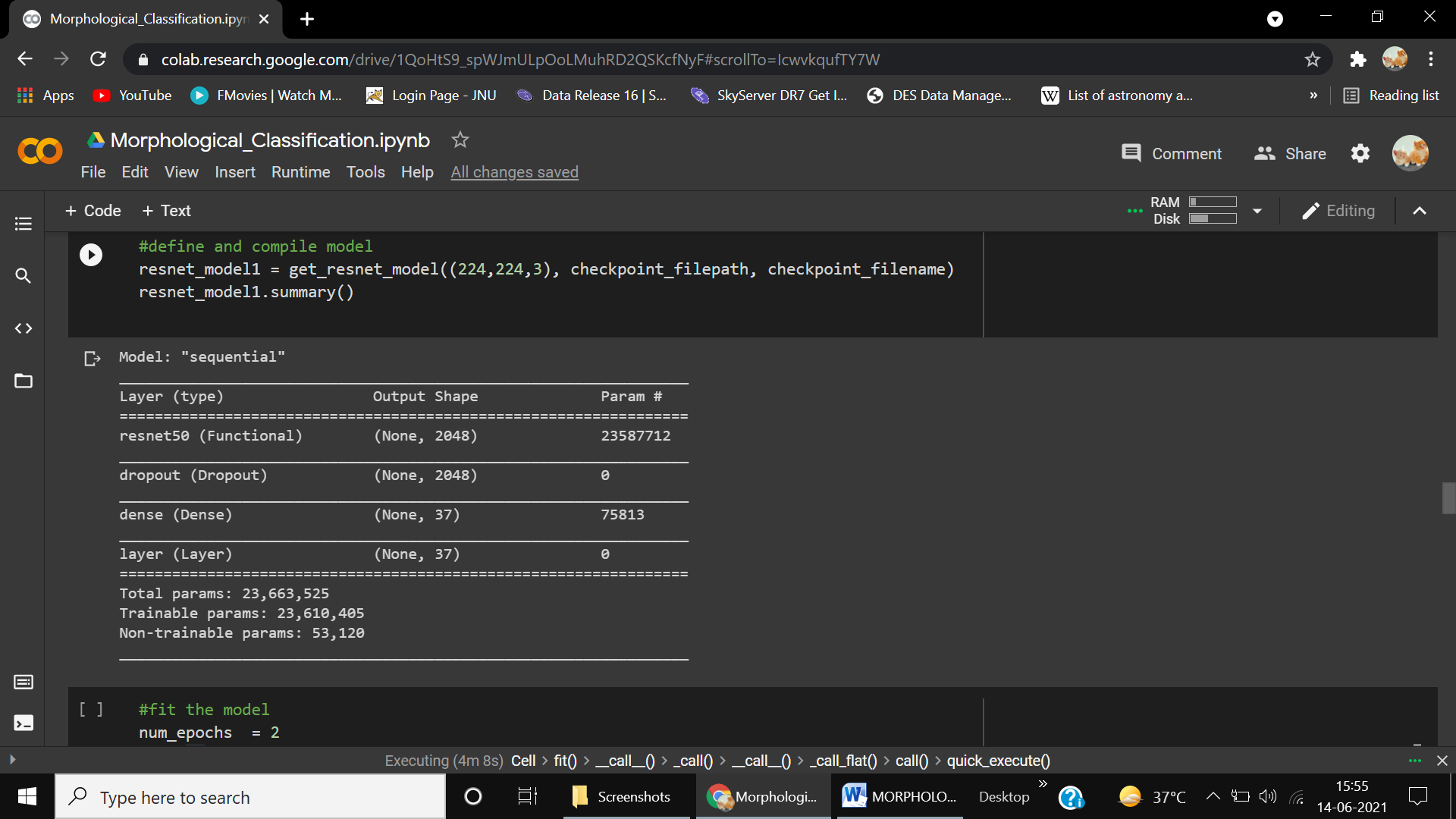
**Figure6:** Customised ReNet-50 architecture

ResNet handles this problem because the residual of the previous information is bypassed to reduce the vanishing gradienteffect. Residuals also help to pass low-level information to high-level layers that make neural networks learn more.

ResNet has many variants e.g., **ResNet**18, **ResNet**34, **ResNet**101, **ResNet**110, **ResNet**-50, **ResNet**-152, etc. I have used ResNet50 for my project. It is particularly an image classifier. However, it also performs well on regression problems. So to better make use of transfer learning [8], this model needs to be modified a little.

There are the following sequential layers in the customized Resnet model

* **Pretrained Resnet without top layers:** First of all, a pretrained Model, with ‘*Imagenet*’ [6] weights, was loaded excluding fully connected top layers. So the average pooling is used so that the last convolutional layer can be connected to the next dense layer.
* **Dense layer 1:**A fully connected dense layer carrying 37 neurons along with 'Relu' activation. This layer is technically an output layer. But, for the different probability distributions in tasks the output of this layer has to be modified. Note that we can’t apply “*Softmax*” or “*Sigmoid*” activation function here as the distribution is not the same for each label. [9]
* **Output Layer:**The final layer of our model contains 37 neurons with explicit customization. Analytically, weights do not need to be initialized with this layer because the performance does seem to increase at all. Hence, I have kept this layer without weight initialization. So basically, it is not a layer, indeed it is an additional 'custom activation function' applied to the previous layer.

****

**Figure7.** Resnet Model summary

1. **Implementation**
   1. **Data Preprocessing**

The very first step in model implementation is data processing. Models do not fit well in the absence of ill-structured data. This is the reason we first clean the data and apply preprocessing techniques if needed before feeding it to the model. Therefore, I have applied the following methods to achieve maximum accuracy and minimum loss.

* + 1. **Crop**

As I have mentioned in the "Analysis section", almost all the galaxies are located in the center of the frame that can be exploited to reduce the size of images. Thus, I have cropped the images so that the resultant images would have size equal to 224×224 pixels.

* + 1. **Resize**

After cropping the galaxies, we need to resize them. Generally, time complexity grows as the number of pixels grows, because now the model needs to process and train more neurons. So it is very crucial to determine an effective size that helps to reduce computation time and does not cause an information loss.

* + 1. **Rescale**

Pixel's value ranges from 0 to 255 which quantify the intensity of that pixel. It is quite clear that data do not lie on the same scale. Hence, scaling is performed to adjust the magnitudes which lie on the same scale. Rescaling converts the scale to (0, 1). Moreover, it also accommodates to reduce training time.



**Resize**

**Crop**



**Figure8.** Cropping and resizing image

* + 1. **Data augmentation**

The biggest problem with fewer data and a bigger model is overfitting. This happens due to the low variance which causes the model to fail. For this, the model does not generalize well and gives poor test accuracy. One of the techniques to overcome this issue is data augmentation. With the help of Image Augmentation more data can be generated. There are the following transformations that can be applied to an image:

* + - 1. **Horizontal Flip**

This function flips the image horizontally. In other words, it generates the mirror image of an original image where the mirror is y-axis.



**Figure9.** A horizontally flipped Image (right) v/s its raw image (left)

* + - 1. **Vertical Flip**

This function flips the image vertically. In other words, it generates the mirror image of an original image where the mirror is x-axis.



**Figure10.** A vertically flipped image (right) v/s its raw image (left)

* + - 1. **Random Rotation**

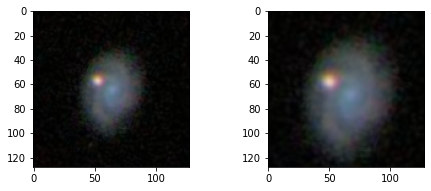
It rotates the image with a random angle within the given range e.g., between 0° to 360°.



**Figure11.** An image with random rotation (right) v/s its raw image (left)

* + - 1. **Zoom**

It zooms in the image by a factor value. This value must be an integer ranging from 0 to 1. Since most of the prominent features of the galaxy lie in the center hence images with zoom-in effect help to find features with high probability.



**Figure12.** An image after zoom-in transformation (right) v/s its raw image (left)

* + - 1. **Brightness**

This function varies the brightness of an image. I have not yet observed whether it increases the performance or not. Hence, I have not used it in the augmentation.



**Figure13.** An image with intensified brightness (between 1 to 2) (right) v/s its raw image (left)

* 1. **Model Compilation**

Both the models are compiled with the following parameters

* **Optimizer:** Adam
* **Loss:** Mean Squared Error
* **Image shape:** The shape of an image
  1. **Model Fitting/Model Training**

To train the model first data is divided into two sets, namely ***train set*** and ***validation set*** in proportion of 80 to 20. Since the data was too large hence loading the data at once was inefficient due to the shortage of memory. Therefore, I have used the data generator to generate the data at run time**.** There are the following important parameters in addition:

* **Training data:** *train\_gen* is the generator that fetches training data (80% of all) as well as processes before loading it.
* **Validation data:** *val\_gen* is the generator that fetches validation data. The proportion of the validation set is 20%.
* **Epochs:** Represents the total number of epochs.
* **Steps per epochs:** Represents the number of iterations per epoch.

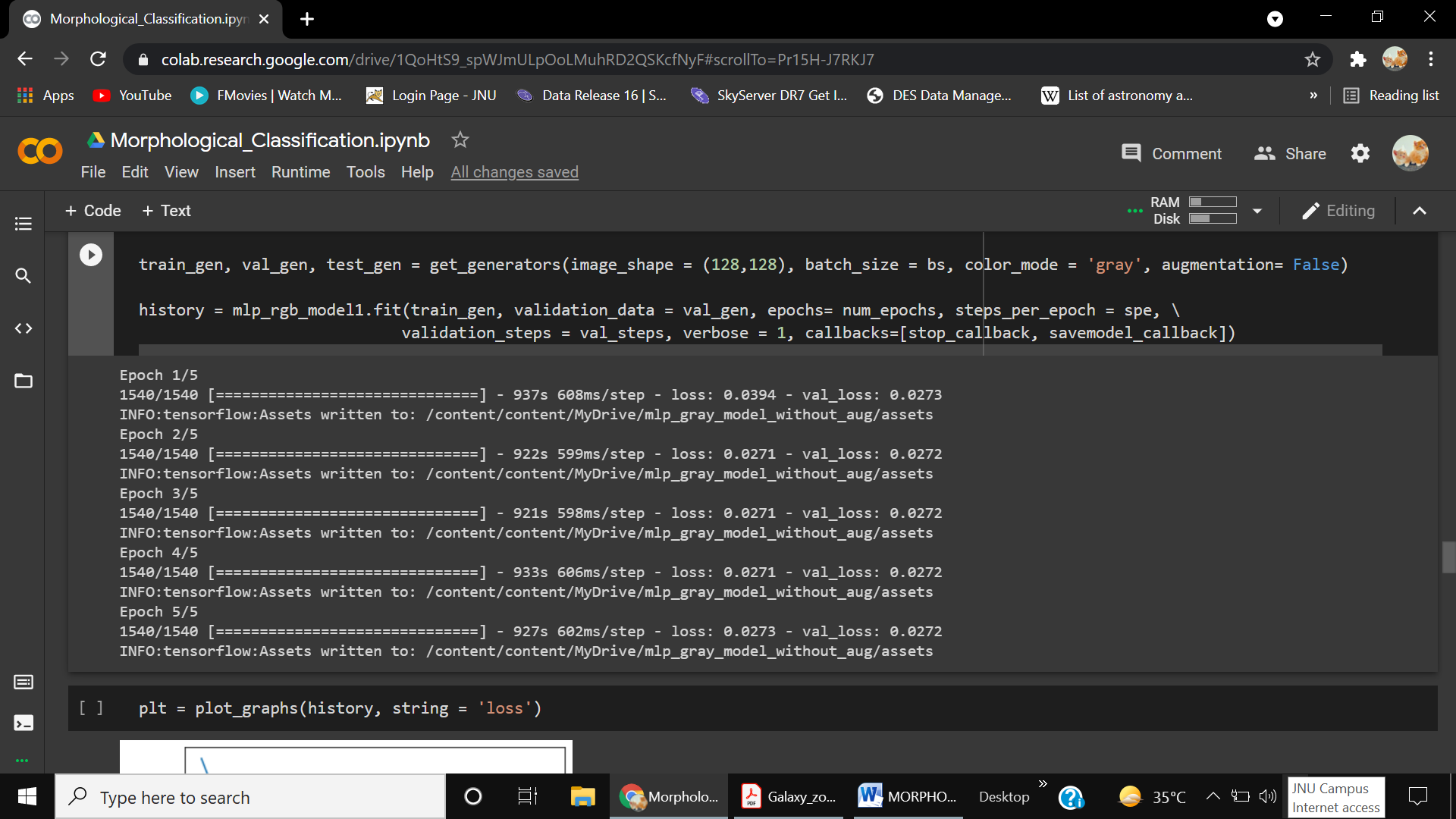
Batch size is 32 in our case.

* **Validation steps:** Number of steps per epochs for validation.
* **Verbose:** It sets the progress bar format.
  + *Verbose 0:* shows nothing in the progress
  + *Verbose 1:* shows the animated progress bar
  + *Verbose 2:* Do not show the bar at all. It only prints the results (loss/accuracy) after the completion of each epoch.
  + **Callbacks:** I have used two Callback functions, namely *"early stopping"*and*"checkpoints"*to save the model. The second Callback helps to save the trained model (weights, architecture, and optimizers' state) so that it can be used next time. It saves a lot of time because training the model every time from scratch is somewhat tedious. These Callbacks run after the end of each epoch.
    1. **MLP Training Analysis**

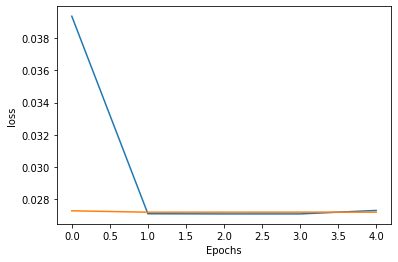
The training is performed with or without data augmentation [9]and with two different modes of image. Finally, results are compared based on the Root Mean Square Error**.**

* + - 1. **Without Data Augmentation**

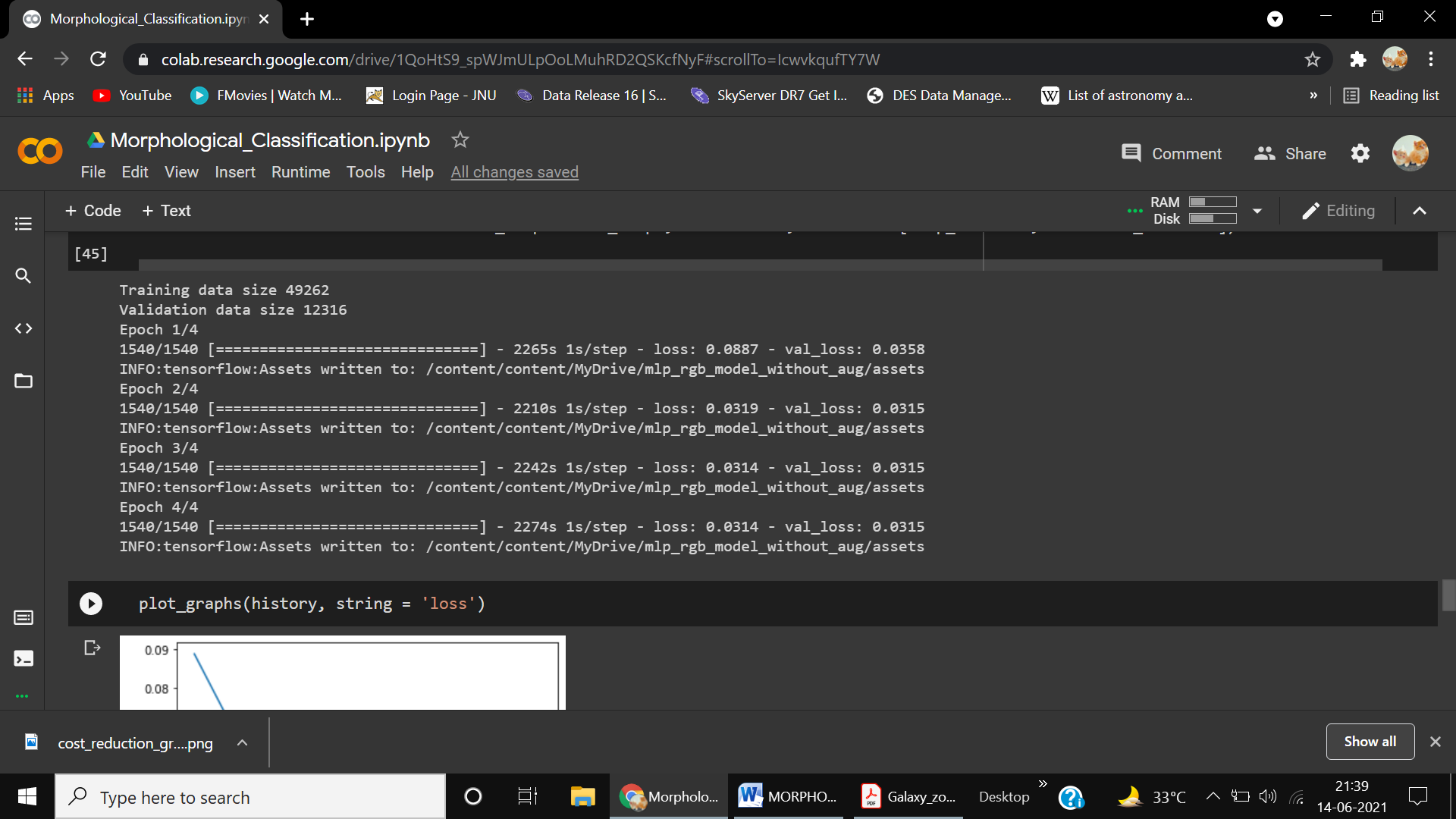
Below figures show the results of training without image augmentation.



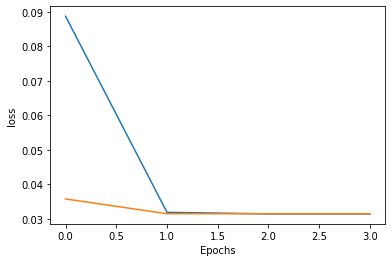
**Figure14.** Training progress up to 5 epochs without data augmentation & Grayscale images



**Figure15** Cost reduction graph for the MLP model with grayscale images and without data augmentation



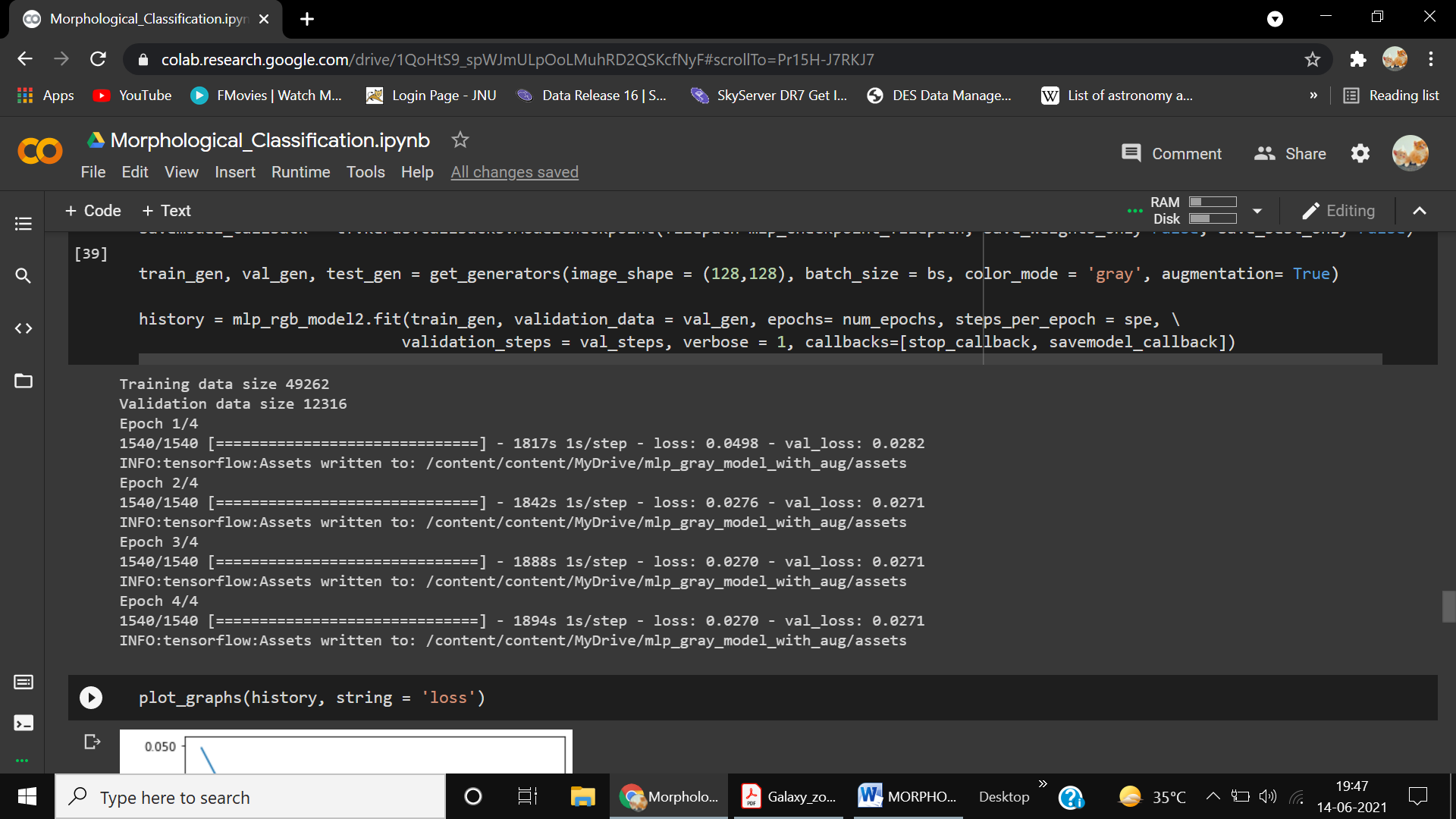
**Figure16.** Training progress up to 4 epochs without data augmentation & RGB images



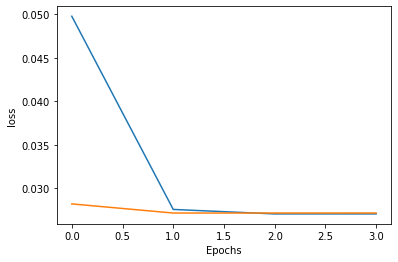
**Figure17.** Cost reduction graph with RGB images and without data augmentation up to 4 epochs.

* + - 1. **With Data Augmentation**

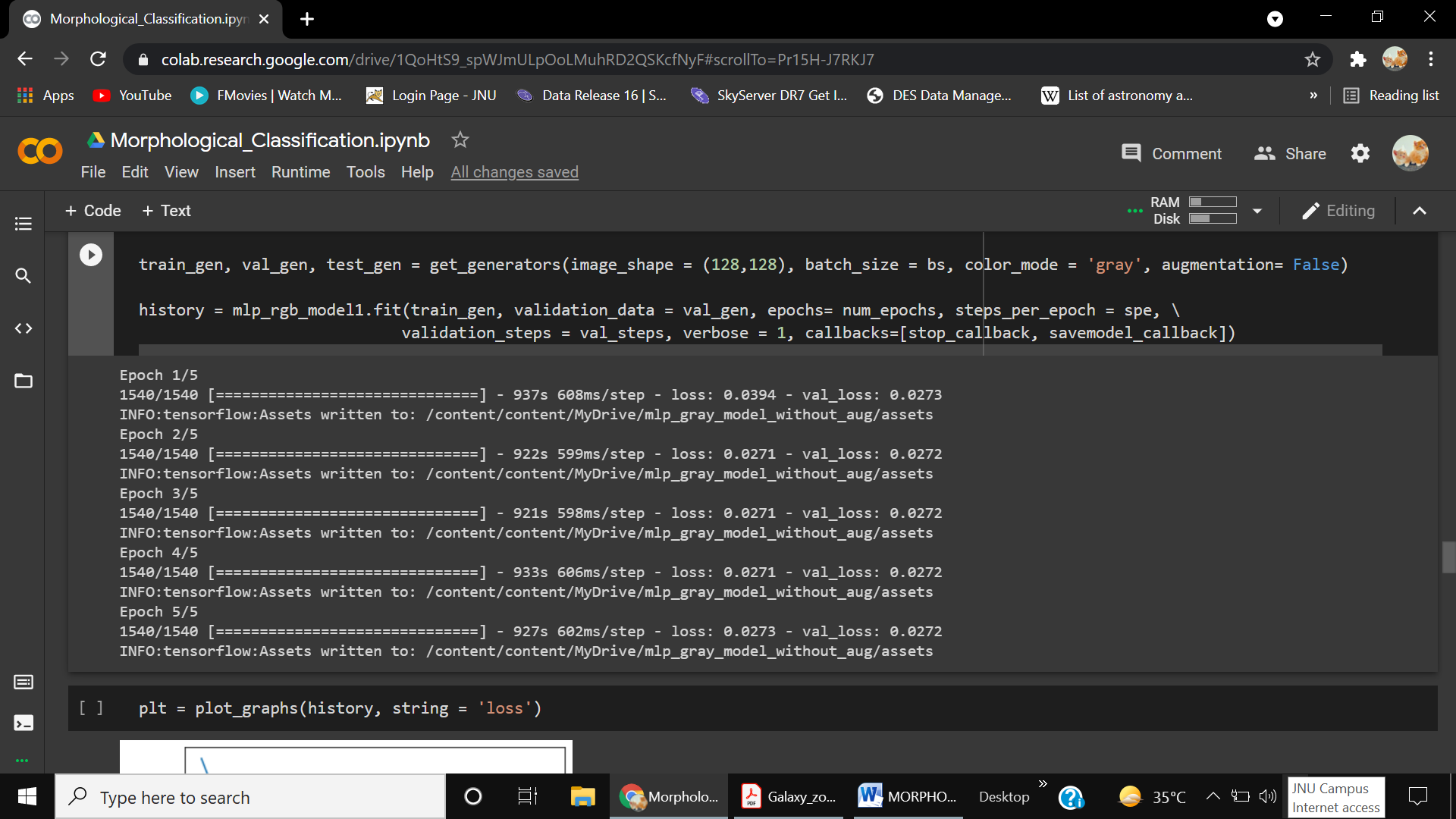
Below figures show the result of training the model with image augmentation.



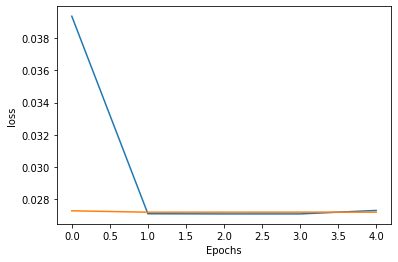
**Figure18.** Training progress up to 4 epochs with data augmentation & Grayscale images



**Figure19.** Cost reduction graph with Gray images and data augmentation.



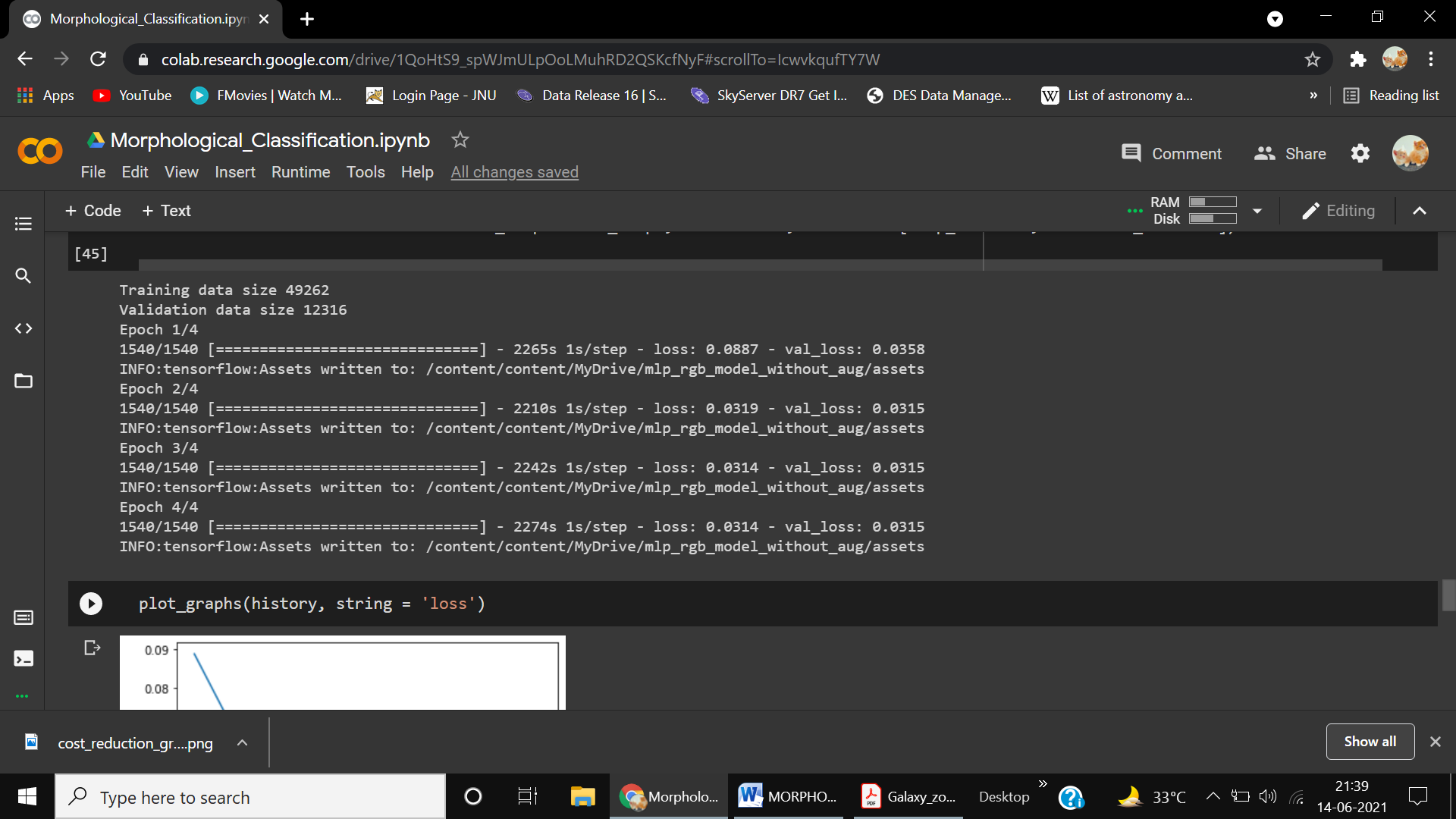
**Figure20.** Training progress (ResNet-50) model up to 4 epochs with data augmentation



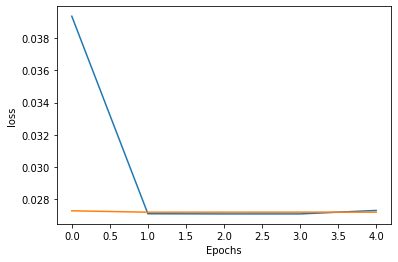
**Figure21.** Cost reduction graph for ResNet-50 model with grayscale images and without data augmentation.

* + 1. **Resnet Training Analysis**
       1. **Without Data Augmentation**

Below figures show the result of training the model without image augmentation.



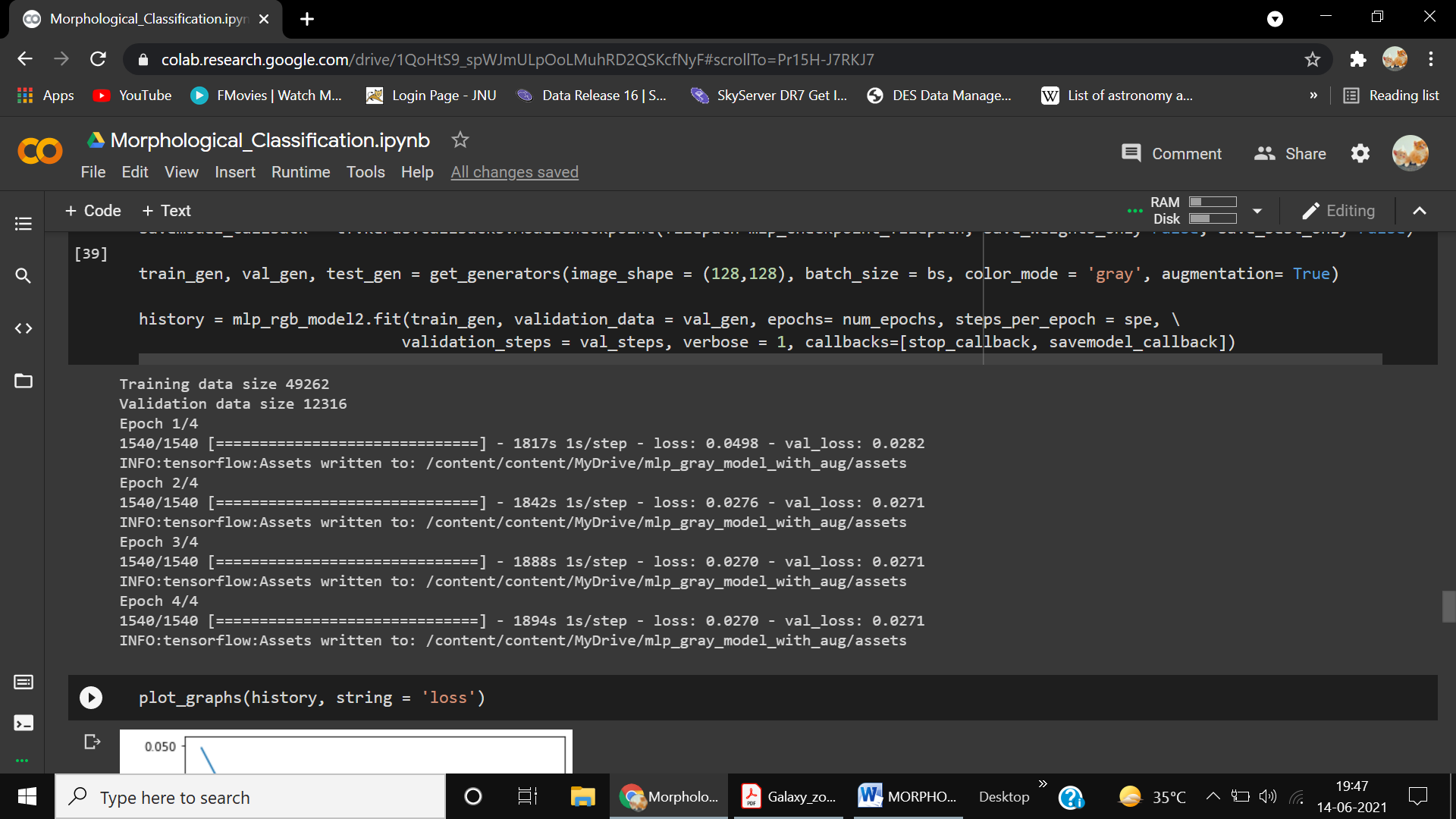
**Figure22.** Training progress up to 4 epochs without data augmentation & RGB images using ResNet-50



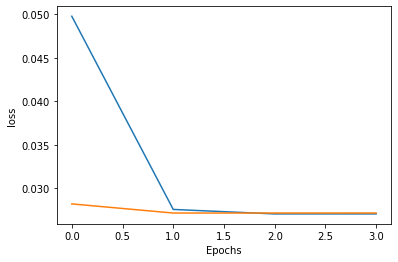
**Figure23.** Cost reduction graph for ResNet-50 model with RGB images and without data augmentation.

* + - 1. **With Data Augmentation**

Below figures represents the result of training the model with image augmentation.



**Figure24.** Training progress up to 4 epochs with data augmentation & RGB images



**Figure25.** Cost reduction graph with RGB images and data augmentation.

1. **Testing**

Testing is performed with a total of 79975 galaxies on the following variants of the model. For which RMSE is shown below:

**Root Mean Squared Error**

RMSE is the standard deviation between actual data and predicted data. It is a statistical measure to compute the loss in either statistical model/machine learning model. When it comes to regression RMSE is the promising loss function to measure how well the regression line is fit on data. To understand it, see the formula given below.

Here,

***i****:* Variable

***N***: Total number of data points or samples in data

: Actual label or observation

: Predicted labelor observation

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **Data Augmentation** | **Image Color Mode** | **RMSE**  **(Root Mean Square Error)** |
| **MLP** | No | Grayscale | 0.0897 |
| **MLP** | Yes | Grayscale | 0.0555 |
| **MLP** | No | RGB | 0.0756 |
| **MLP** | Yes | RGB | 0.0498 |
| **Resnet50** | No | RGB | 0.0450 |
| **Resnet50** | Yes | RGB | 0.0289 |

**Table3.** RMSE for different version of the models

1. **Conclusion and Future Work**
   1. **Conclusion**

The whole analysis concludes that Resnet50 Model surpassed MLP model and 3 channel images produced better accuracy than single-channel images. It is important to note that for grayscale images training gets saturated after few epochs but for RGB images it continues to slow down. It is because 3 channel images contain more information than single-channel images. Indeed, color holds huge importance in the evolution of galaxies. It is challenging to extract the complete features of the galaxies, however, most of the prominent features of a galaxy lie in its center and so is its deep history. Therefore, the center of a galaxy must be observed thoroughly to extract maximum features out of that. Though astronomers are regularly trying to write better and better algorithms to unveil the complete history of our mesmerising universe, there is still a long way to go. But it is worth remembering that "The Universe has a way of leading you where you are supposed to be at the moment you are supposed to be there". Hence, never stop searching, discovering, and learning.

* 1. **Future Scope**

This classification can also be extended to a large scale. The model can be trained with GZ2 data and can be tested on DES data release 2 (ref). The advantage of selecting DES data is high-resolution images. These images comprising of (10000, 10000) pixels and contain millions of objects altogether. This classification can help to assert the existence of dark energy and dark matter.

1. **Bibliography/References**

|  |  |
| --- | --- |
| [1] | T. M. C. ABBOTT, "DES Collaboration," *ApJS,* vol. 235, no. 3, p. 30, 2020. |
| [2] | Willett, "Galaxy zoo 2: detailed morphological classifications for 304,122 galaxies from the Sloan Digital Sky Survey," *MNRAS,* vol. 435, no. 4, pp. 1-29, 2013. |
| [3] | Lintott, "Galaxy zoo 1: Data Release of Morphological Classifications for nearly 900,000 galaxies," *MNRAS,* vol. 410, no. 1, p. 14, 2008. |
| [4] | D. G. York, "THE SLOAN DIGITAL SKY SURVEY: TECHNICAL SUMMARY," *THE ASTRONOMICAL JOURNAL,* vol. 120, no. 3, p. 9, 2000. |
| [5] | K. He, X. Zhang, S. Ren and S. Jian, "Deep Residual Learning for Image Recognition," *Proceedings of the IEEE conference on computer vision and pattern recognition.,* vol. 1, pp. 770-778, 2016. |
| [6] | J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," *2009 IEEE Conference on Computer Vision and Pattern Recognition,* p. 8, 2009. |
| [7] | Albawi, S. a. Mohammed, T. A. a. Al-Zawi and Saad, "Understanding of a convolutional neural network," *2017 International Conference on Engineering and Technology (ICET),* pp. 1-6, 2017. |
| [8] | K. a. K. T. M. a. W. D. Weiss, "A survey of transfer learning," *Journal of Big data,* vol. 3, no. 1, pp. 1-40, 2016. |
| [9] | X.-L. M. David A van Dyk, "The Art of Data Augmentation," *Journal of Computational and Graphical Statistics ,* vol. 10, no. 1, pp. 1-50, 2001. |

1. <https://www.kaggle.com/c/galaxy-zoo-the-galaxy-challenge/> [↑](#footnote-ref-1)
2. <https://www.darkenergysurvey.org> [↑](#footnote-ref-2)
3. <https://www.sdss.org/> [↑](#footnote-ref-3)
4. <https://solarsystem.nasa.gov/resources/2319/first-image-of-a-black-hole/> [↑](#footnote-ref-4)
5. <https://www.lsst.org/> [↑](#footnote-ref-5)
6. <https://www.kaggle.com/c/galaxy-zoo-the-galaxy-challenge/> [↑](#footnote-ref-6)