**0. Abstract**

Galaxy morphology classification and detection is the process of identifying the shape and structure of a galaxy based on its appearance. This is an important task in the field of astronomy, as the shape and structure of a galaxy can provide clues about its history, evolution, and contents. For example, spiral galaxies are known to contain both old and young stars, as well as gas and dust, which allows them to form new stars, while elliptical galaxies are made up of mostly older stars and have relatively little gas and dust.

One approach to galaxy morphology classification is to use machine learning techniques, such as convolutional neural networks (CNNs). CNNs are a type of artificial neural network that are well-suited to image classification tasks, as they are able to automatically learn and extract features from images. Our next goal after using CNNs is to make our model practical and hardware friendly so it can have real world applications.

1. **INTRODUCTION**

Astronomer Edwin Hubble **pioneered** the **classification** **of** celestial giants called galaxies. Galaxies are **gravitationally** **bound** **systems** of stars, interstellar gas, dark matter, dust, and **star** **debris.** Galaxies form over billions of years, and their morphology **(essentially** their shape and general **appearance)** gives astronomers **a** **lot** **of** information about their composition and evolution.

**Astrophysicists** use large catalogs of information to test existing theories **and** **make** new conjectures to explain the physical processes **that** **drive** galaxies, star formation, and the nature of the universe. **Classification** **of** **galaxies** **is** **important** **because** **there** **are** **many** .Currently, astronomers manually classify galaxies based on visual inspection of photographs. This method is **time** **consuming** and not a **worthwhile** activity for **astronomers.** This method is also prone to human **error,** **leading** **to** inaccuracies and misclassifications. Astronomy has seen **a** **recent** explosion of **data** **with** programs **such** **as** the Sloan Digital Sky Survey **(SDSS).**

The Sloan Digital Sky Survey (SDSS), **launched** in 2000, collected **the** **most** data **in** **the** **history** **of** **astronomy** in its first few **weeks.** **Twenty** years later, **the** **article** published **in** EDP Sciences **A122** contains about 170 terabytes of **his** information. Soon its successor, the Large Synoptic Survey Telescope (LSST), will **collect** **this** **amount** of data every five days (York et al. 2000). It provided **an** entry **point** for computer scientists **seeking** to engage in astronomical research, and **explained** why big data mining and machine learning methods are gaining **acclaim.**

**Access** to this amount of data has only **been** possible in the **last** decade, **so** **computational** **sky** classification is a very **advanced** **area** with **plenty** **of** **potential** for machine learning and image processing **applications.** Our goal is to apply machine learning algorithms to the **iterative** task of galaxy classification on **huge** data **sets.** This not only **reduces** classification **errors,** but also **allows** astronomers to pursue more **exciting** tasks. At the same time, we tried to **provide** **a** **model** **with** **fewer** parameters and **usable** for mobile and web applications.

**1.1 Convolutional Neural Networks**

Most computer vision algorithms use something called a **convolutional** neural **network** **(CNN).** A CNN is a model used in machine learning to extract **features** **such** **as** **textures** and **edges** from spatial data.

Like basic **forward** neural networks, CNNs learn from their inputs and adjust **the** parameters (weights and biases) to make accurate predictions. **What** makes **CNN** **special,** **however,** is **its** ability to extract features from images. CNN **processes** images as **a** **matrix** and can extract spatial features such as texture, edges, and depth. This is done using **convolution** and **compositing** **layers.** We can think a convolutional layer of as a series of feature maps. A **convolution** layer applies a set of image filters to an input image represented as a matrix. The resulting filtered image or feature map looks different because it extracted distinct features. These image filters are termed convolution kernels.

1. **Review of State-of-Art and Related Works**

**Within** the **framework** of **the** **taxonomy** and **morphology** of **galaxies,** we **consider** below **a** **number** **of** works **that** **have** **developed** different approaches and **have** **made** great efforts to **define** the morphological types of **galaxies.** **ha** from SDSS in visual and **automatic** **mode.**

**Callejaet.** **Al** **published** an experimental study using machine learning and image analysis to classify galaxies. A neural network with a locally weighted regression **method** and **a** **homogeneous** **set** of classifiers **was** used. They used the bagging ensemble method for neural networks and manipulated input features to create locally weighted **regression** **ensembles.** The **galaxies were** **fully** **automatically** transformed by **rotation** and **center** **clipping.** **In** **addition,** **principal** **component** **analysis** (PCA) was used **to** **reduce** **the** **size** and extract relevant information from the image data. The preliminary **test** results were evaluated **using** **the** **10-fold** **cross-validation** **technique** and it **was** **shown** that the **homogeneous** locally weighted regression **set** **produced** the best results, with **an** accuracy **of** **91%.** when considering three types of galaxies **(elliptical,** **spiral** and **irregular)** and **95%** when considering two types **(elliptical** and **spiral).**

Kasivajhula et al. (2007) **investigated** **support** **vector** **machines,** random **forests,** naive Bayes algorithms as galaxy image classifiers, and principal component analysis for direct **compression** **of** image pixel **data,** but **gave** **preference** **to** random **forests.** Nonetheless, Andrae et al. (2010) applied probabilistic classification to classify **bright** SDSS galaxies and **found** that it produces **acceptable** morphological classes and object-to-class assignments without **previous** assumptions.

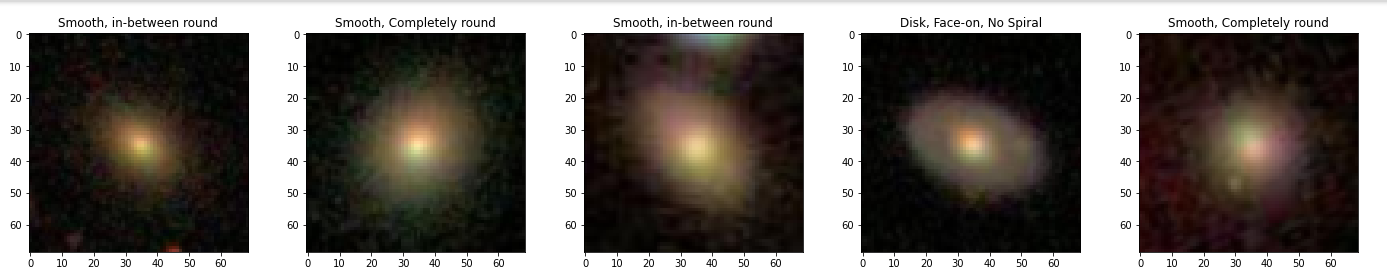
**KChou** et. **Al.** set out to **develop** an algorithm that can **abstract** indicators **of** galaxy **morphology.** He developed a pipeline **that** **combines** multiple computer vision feature detectors **with** ML regression. **Performance** **was** **tested** **using** **cross-validation** **techniques.** The **pipeline** **has** **three** **stages: Feature** extraction, machine learning regression, probability normalization. **Several** techniques **are** used for image analysis, **such** **as** PCA, SIFT, **Hog** **and** Fourier **transform.** **Its** neural network is **pretrained** called **Overfeat** trained on the ImageNet **dataset.**

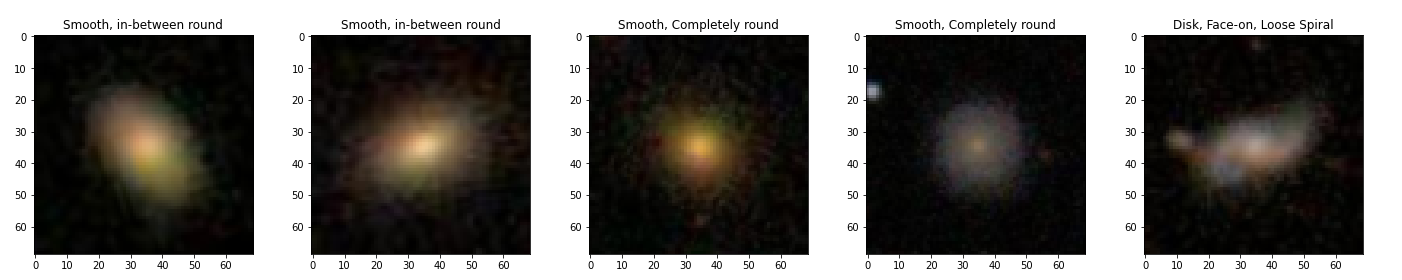
**Gonzales et al. Al. conferred a method for automatic galaxy detection and classification, including a new extension procedure to make the trained model more robust to data from different instruments and contrast stretch functions. He trained deep learning models using public data such as SDSS and Galaxy Zoo datasets and private data such as Next Generation Virgo (NGVS) and Fornax (NGFS) surveys. The training was strongly tied to his conversion method from FITS raw data to his 3-channel RGB images. Therefore, it is recommended to use 5 transformation methods in data augmentation. This greatly improved the overall detection of galaxies by various instruments, data reduction methods and bands. We trained a detection and classification method using the deep learning framework DARKNET and the real-time object detection system YOLO. They were implemented in C and CUDA platforms and made extensive use of graphics processing units (GPUs) capable of processing SDSS images in 50 ms or DECam images in about 3 seconds.**

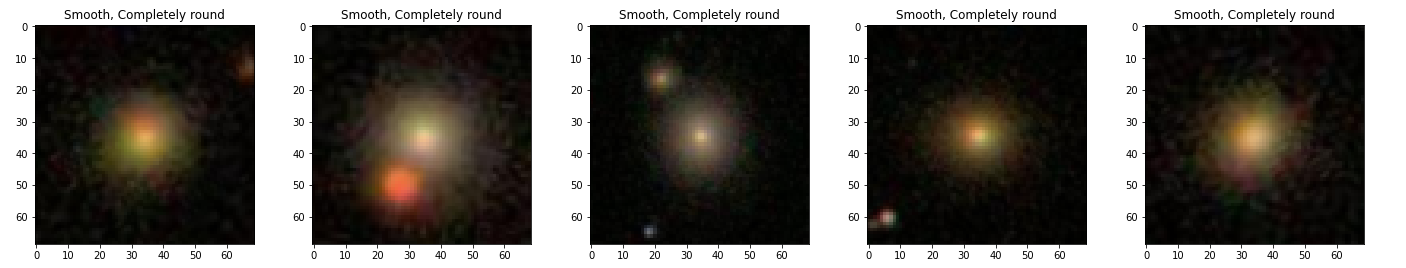
**DATASET**

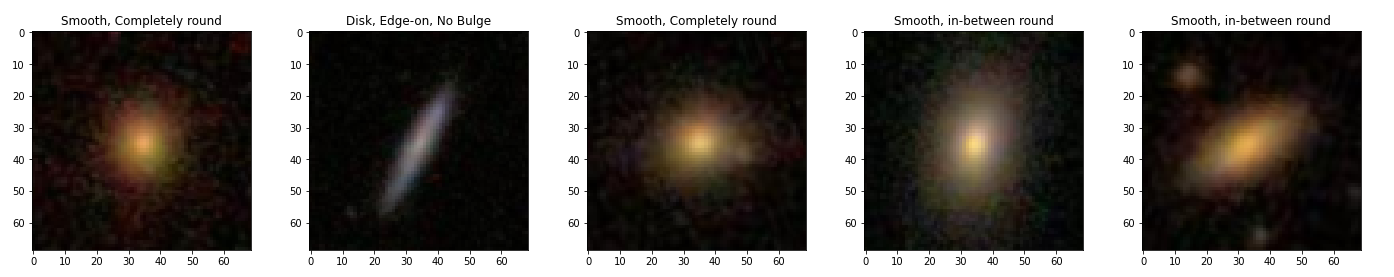
For our study, we used AstroNN’s SDSS dataset which contains 21785 69x69 pixels colored galaxy images separated in 10 classes. The name of the classes along with their example images is showcased in fig1.

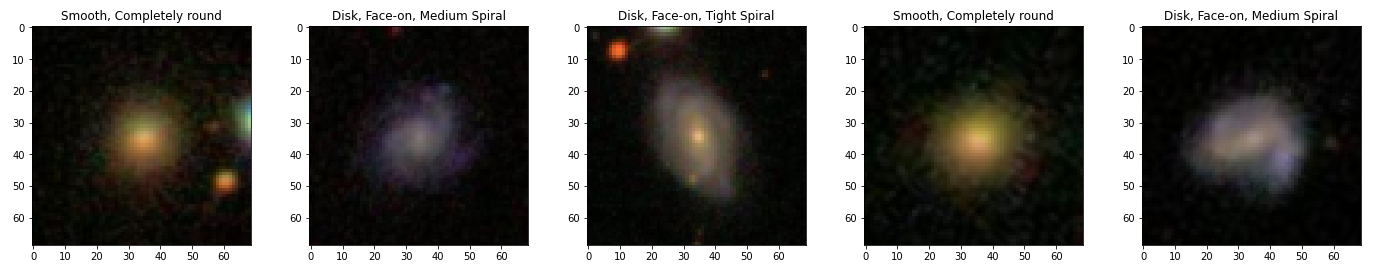
**Fig1.**

****

****

****

****

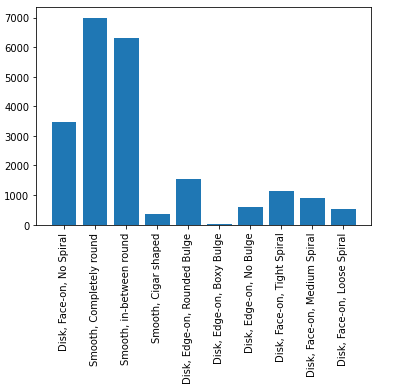
****

All the label information for this dataset was taken from GalaxyZoo. Galaxy Zoo is a project that relies on volunteers to classify galaxy images, and as a result, there can be disagreements among the volunteers. To address this, Galaxy10 only includes images where more than 55% of the votes agree on the classification, resulting in a dataset of 21,785 images. This threshold was chosen based on validation and is meant to be an alternative to MNIST or Cifar10 as a deep learning toy dataset for astronomers. The images were cropped and downscaled to make them manageable on most computer and graphics card memory

The class distribution of these images is given in Table2.

Table 2.

| Class No. | Class Name | No. of Images |
| --- | --- | --- |
| 1 | Disk, Face-on, No spiral | 3461 |
| 2 | Smooth, Completely round | 6997 |
| 3 | Smooth, In-between round | 6292 |
| 4 | Smooth, Cigar shaped | 394 |
| 5 | Disk, Edge-on, Rounded bulge | 1534 |
| 6 | Disk, Edge-on, Boxy bulge | 17 |
| 7 | Disk, Edge-on, No bulge | 589 |
| 8 | Disk, Face-on, Tight spiral | 1121 |

****

1. **Proposed Solution(s)**

In order to make our work application friendly, we tried solving our problem statement with the aim of making our model as smaller as possible. Therefore, we implemented small CNN’s namely Squeezenet and MobileNet which gives Alex-net level accuracy with fewer parameters.

**Squeezenet**

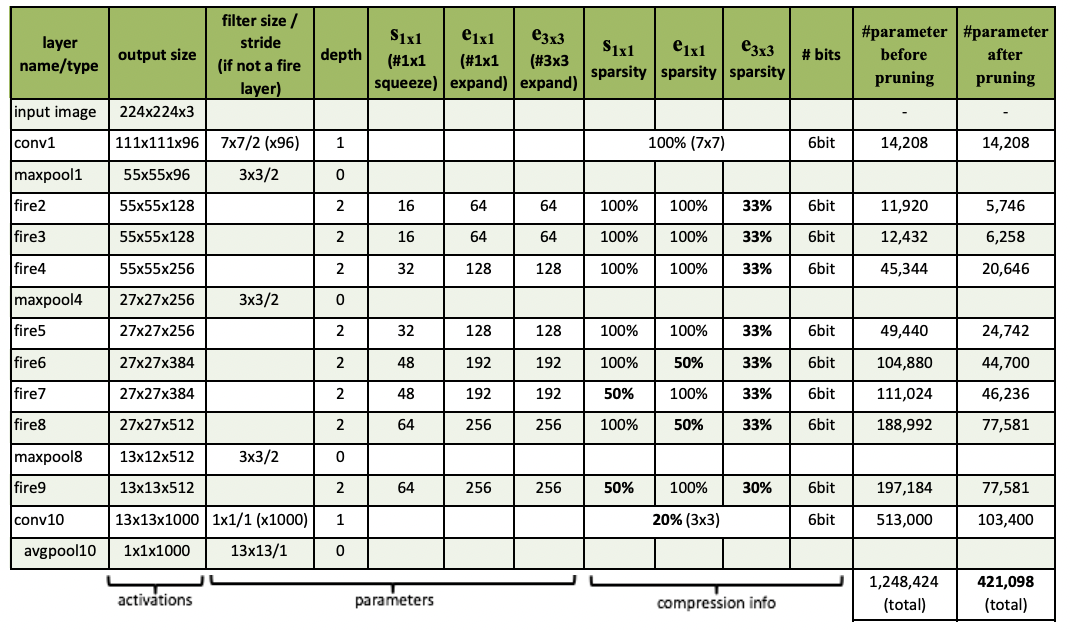
SqueezeNet is a neural network architecture designed for efficient deep learning on constrained computational resources, such as mobile devices or embedded systems. It achieves this by using a combination of techniques such as network pruning, reduction in the number of input channels, and the use of 1x1 convolutional filters to reduce the number of parameters and operations required.

The core of the SqueezeNet architecture is the Fire module, which consists of a squeeze layer that uses 1x1 convolutions to reduce the number of input channels, followed by an expand layer that uses a combination of 1x1 and 3x3 convolutions to generate a set of output feature maps. The Fire module is repeated multiple times to form the overall network.

SqueezeNet also uses techniques such as global average pooling and softmax regression for classification, which further reduce the number of parameters in the network. Overall, SqueezeNet achieves state-of-the-art accuracy on several image classification benchmarks while being much more computationally efficient than other popular neural network architectures.

SqueezeNet begins with a standalone convolution layer, followed by 8 Fire modules and then ending with a final conv layer. There is a gradual increase in the number of filters per fire module from the beginning to the end of the network. SqueezeNet performs max-pooling with a stride of 2 after layers conv1, fire4, fire8, and conv10; these relatively late placements of pooling are per Squeezenet’s Strategy. The full SqueezeNet architecture is presented in Table 1.

**Table 1.**



**Methodology**

* **Data loading and pre-processing**

For our study, we have used the galaxy10 dataset from a python module called AstroNN. Images and Labels are loaded from the respective module where further processing like normalizing the images and hot encoding the labels is done. We further split the data into training and testing at a ratio of 85 : 15.

* **Architecture construction**

In our model, the fire module is the most crucial component.

The Fire module has been widely adopted in various deep learning architectures, not just in SqueezeNet, but also in other models such as MobileNet and EfficientNet. Its ability to reduce model size and computational requirements has made it a popular choice for edge devices and other resource-constrained platforms. The structure of the fire module is shown in Fig2.

We have used convolution layers to build this fire module. The activation function “relu” is used in these convolution layers. Rectified Linear Unit (relu) function returns 0 if it receives any negative input, but for any positive value it returns that value back. So it can be written as :

f(x)=max(0,x).

The filter is applied using convolution layers to produce a feature map.

Each 2D convolution layer uses a n x n array and a m x m kernel to apply k

filters and retrieve k features. Here, the input image's (individually the r, g, and b bands) dimensions are n x n, the convolution kernel's dimensions are m x m,

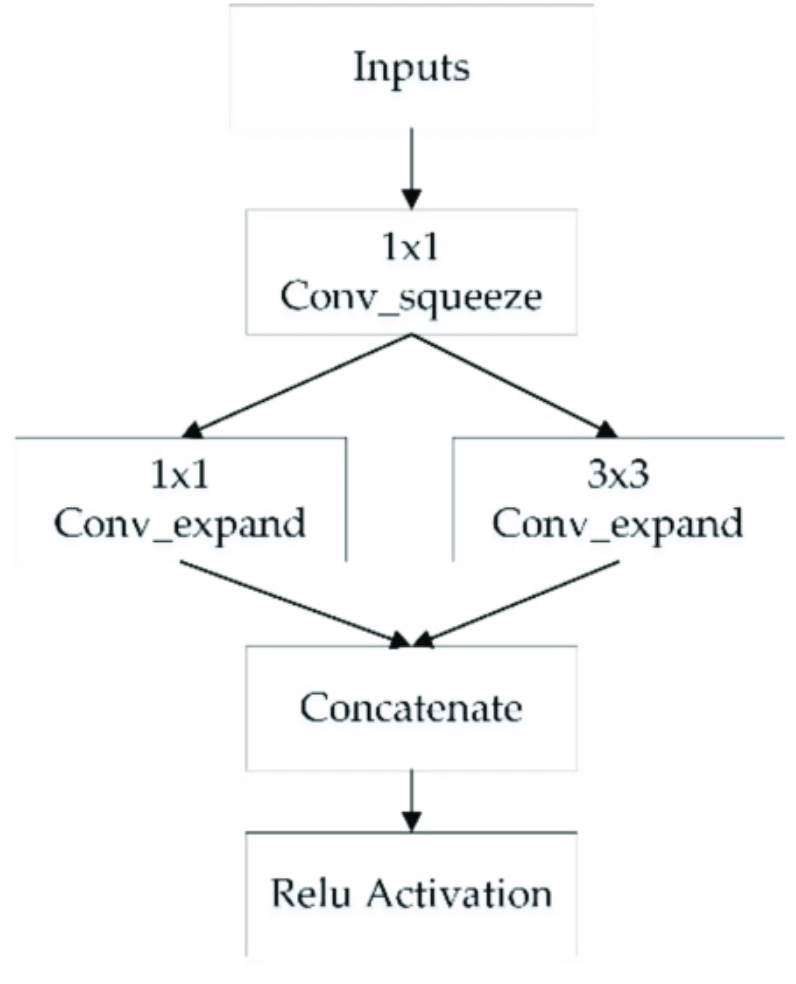
and the number of filters is k.

Our model architecture has multiple fire module blocks followed by a few Maxpooling layers in between those fire blocks. Similarly, Global average pooling is applied at the end of our architecture.

We employ pooling layers with convolution layers to reduce the feature map's sample size. In this study, MaxPooling was employed to assist keep the maximum components and lessen the noise and GlobalAverage pooling was employed to generate one feature map for each corresponding category of the classification task in the last convolution layer.

"Softmax" serves as our activation function in the output layer. This is done so that the model's output will be the likelihood that a picture belongs to a certain class.

This implementation of Squeezenet architecture was done using Keras tensorflow.

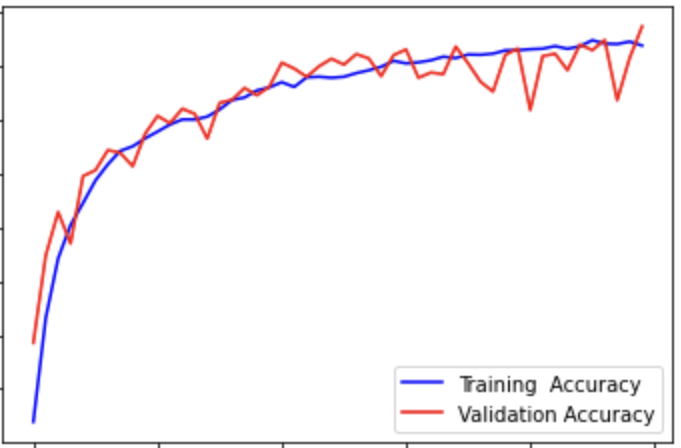


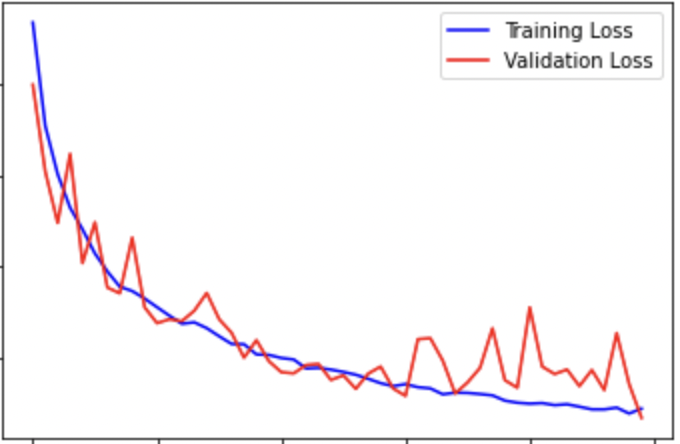
**Fig 2.** Structure of the fire module

* **Model training**

As we split the data into training and testing at the ratio of 85 : 15, the model training was done on 18517 images and testing on 3268 images respectively. The total number of parameters shown in the model summary were 740,554. While training, Categorical cross-entropy loss was used as the loss function and ‘adamax' as the optimizer. Model training was done over 100 epochs where maximum training accuracy of 99.5% was achieved. The training and validation, accuracy and loss curve graphs are displayed below.

**Fig 3.** Accuracy and Loss curves.



****

**MobileNet**

MobileNet is **an** **open** **source** computer vision model **from** Google and **is** designed **to** **train** classifiers. It uses **deep** **convolutions** to **greatly** reduce the number of parameters compared to other networks, resulting in a **mildly** deep neural network. MobileNet is Tensorflow's first mobile computer vision model.

It is a lightweight deep neural **network.**

A **depth-separable** convolution is **obtained** from two operations:

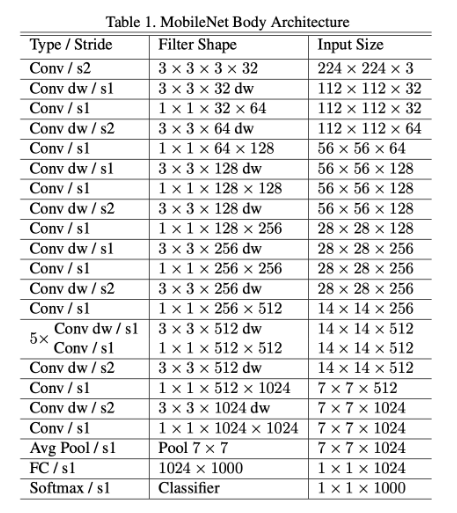
**1..** **Point** convolution

2. **Deep** convolution

**Network** speed and power consumption **are** proportional to the **multiplier-accumulator** **(MAC),** which measures the number of multiplication and addition **operations** **merged.**

ARCHITECTURE:

The MobileNet model is based on depthwise separable **convolutions.** **This** is a form of **factored** **convolution** **that** **factors** **the** standard convolution into depthwise **convolutions** and **1** **×** **1** **convolutions** called pointwise **convolutions.** For **MobileNet,** **depth** convolution applies **one** filter to each input channel. The pointwise convolution applies a 1×1 convolution **and** **combines** the **output** with the depthwise convolution. A standard convolution filters **the** **inputs** and combines **them** into a new set of outputs in one step. **Depthwise** separable convolution splits this into two **layers.** **Another** layer for filtering and **another** layer for **merging.** This factorization has the effect of **significantly** reducing computation and model size.



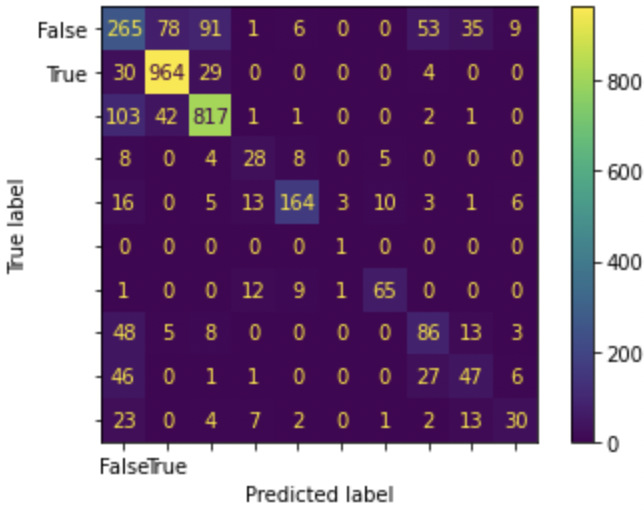
1. **Experimental Results:**

In this study, we implemented SqueezeNet and MobbileNet on AstroNN’s SDSS dataset for classifying Galaxies. The purpose of this study was to evaluate the performance of these models on this dataset and compare its accuracy with other state-of-the-art deep learning models.

**SqueezeNet:**

We evaluated the performance of SqueezeNet on the test set and achieved a training accuracy of 99.5% and an overall test accuracy of 80%. The confusion matrix and classification report for the classification results is shown below.

**Fig4.** Confusion Matrix for Squeezenet



**Table 3.** Classification Report

|  | Precision | Recall | F1-score | Support |
| --- | --- | --- | --- | --- |
| 0 | 0.61 | 0.63 | 0.62 | 536 |
| 1 | 0.90 | 0.91 | 0.91 | 1029 |
| 2 | 0.88 | 0.88 | 0.88 | 972 |
| 3 | 0.71 | 0.44 | 0.54 | 50 |
| 4 | 0.85 | 0.88 | 0.86 | 225 |
| 5 | 0.00 | 0.00 | 0.00 | 1 |
| 6 | 0.87 | 0.85 | 0.86 | 103 |
| 7 | 0.54 | 0.53 | 0.54 | 148 |
| 8 | 0.52 | 0.50 | 0.51 | 133 |
| 9 | 0.57 | 0.61 | 0.59 | 71 |
|  |  |  |  |  |
| Accuracy |  |  | 0.80 | 3268 |
| Macro Avg | 0.65 | 0.62 | 0.63 | 3268 |
| Weighted Avg | 0.80 | 0.80 | 0.80 | 3268 |

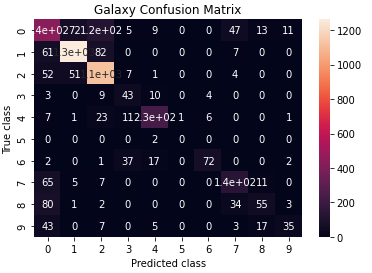
**MobileNet:**

We evaluated the performance of MobileNet on the test set and achieved for the test data(50 epochs ) and train data as follows :

Train data Accuracy : 96.79%

Test data Accuracy : 78%

The confusion matrix for the classification of galaxy using MobilNet is as follows:



**DISCUSSION**

While achieving a test accuracy of 82.4% with CNN architecture is certainly commendable, there is always room for improvement. One potential area of improvement is the use of Squeezenet architecture, which is known for its efficiency and low parameter count. However, the trade-off for such a lightweight architecture is often lower accuracy. Therefore, it is important to carefully balance the need for accuracy with the limitations of the architecture or the hardware on which the model will be deployed.

In the case of our research project, deploying the model on a website or mobile app as a lightweight model is a feasible solution. Although the accuracy may be lower, the benefits of making the model accessible to citizen scientists and increasing the number of people who can contribute to galaxy classification is significant. This approach is especially valuable for large-scale citizen science initiatives and community-driven projects. Overall, by using efficient architecture and deploying models on lightweight platforms, we can not only improve the accuracy of classification but also engage a broader audience in the scientific process.

**5.Conclusion and future scope:**

Deploying a galaxy morphology classification model on a web or mobile app can open up several exciting possibilities for future scope and user contributions. Here are some potential avenues to explore:

1. Increased User Participation: By making the classification model available on a user-friendly platform, such as a web or mobile app, more people will be able to contribute to the project. This could greatly increase the amount of data that can be classified, which in turn could lead to a more accurate and comprehensive model.
2. Real-Time Classification: Edge computing can be used to classify galaxy morphology in real-time. By deploying the model on mobile devices or on the edge of the network, users can contribute to the classification process even when they are offline. This can greatly increase the amount of data that can be classified, and it can also reduce the computational burden on the central server.
3. Improved Model Accuracy: With more data and more user input, the model can be trained to be more accurate over time. As users classify more galaxies and provide feedback on the accuracy of the model's predictions, the model can be fine-tuned to improve its performance.
4. Educational Opportunities: Deploying the model on a web or mobile app can provide an opportunity to educate the public about galaxies and their morphology. By providing a user-friendly interface and educational resources, users can learn about the science behind galaxy classification while contributing to the project.
5. New Research Directions: By making the classification model available to a wide range of users, new research questions may emerge. For example, researchers may be able to use the data to study the distribution of galaxy morphology in different regions of space or to investigate the relationship between galaxy morphology and other astronomical phenomena.

**Conclusion**

In conclusion, the use of efficient architectures such as Squeezenet can be a valuable tool for improving the accessibility and accuracy of classification models, especially in citizen science initiatives and community-driven projects. While there may be trade-offs in terms of accuracy, the benefits of engaging a broader audience in the scientific process and increasing the number of people who can contribute to galaxy classification are significant. By carefully balancing the need for accuracy with the limitations of the architecture or the hardware on which the model will be deployed, we can make significant strides in improving the efficiency and accessibility of classification models. Overall, the integration of lightweight models and efficient architectures has the potential to democratize the scientific process and bring us closer to a more comprehensive understanding of the universe.

**6. REFERENCES**

1. [Forrest N. Iandola](https://arxiv.org/search/cs?searchtype=author&query=Iandola%2C+F+N), [Song Han](https://arxiv.org/search/cs?searchtype=author&query=Han%2C+S), [Matthew W. Moskewicz](https://arxiv.org/search/cs?searchtype=author&query=Moskewicz%2C+M+W), [Khalid Ashraf](https://arxiv.org/search/cs?searchtype=author&query=Ashraf%2C+K), [William J. Dally](https://arxiv.org/search/cs?searchtype=author&query=Dally%2C+W+J), [Kurt Keutzer](https://arxiv.org/search/cs?searchtype=author&query=Keutzer%2C+K). SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size. arXiv:1602.07360[cs.CV].

[With the help of this research paper, we were able to successfully implement the SqueezeNet architecture for Galaxy detection].

1. Lintott, Chris J. and Schawinski, Kevin and Slosar, Anvze and Land, Kate and Bamford, Steven and Thomas, Daniel and Raddick, M. Jordan and Nichol, Robert C. and Szalay, Alex and Andreescu, Dan and Murray, Phil and Vandenberg, Jan. Galaxy Zoo: morphologies derived from visual inspection of galaxies from the Sloan Digital Sky Survey. arXiv:10.1111/j.1365-2966.2008.13689.x.

[The data and classification information used by us is extracted from this research paper].

1. Lintott, Chris ; Schawinski, Kevin search by orcid ; Bamford, Steven search by orcid ; Slosar, Anå¾e ; Land, Kate ; Thomas, Daniel search by orcid ; Edmondson, Edd ; Masters, Karen search by orcid ; Nichol, Robert C. ; Raddick, M. Jordan ; Szalay, Alex search by orcid ; Andreescu, Dan ; Murray, Phil ; Vandenberg, Jan. Galaxy Zoo 1: data release of morphological classifications for nearly 900 000 galaxies. arXiv:1007.3265.

[The data and classification information used by us is extracted from this research paper].

1. Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, Hartwig Adam. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv:1704.04861 [cs.CV].

[With the help of this research paper, we were able to successfully implement the MobileNet architecture for Galaxy detection].

**7. Gantt Chart**

| **Tasks** | **Start** | **End** | **Dec 22** | | | | **Jan23** | | | | **Feb23** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Research paper’s study on Galaxy detection using Deep learning** | **29/11/22** | **5/12/22** | **W1** | **W2** | **W3** | **W4** | **W5** | **W6** | **W7** | **W8** | **W9** |
|  |  |  |  |  |  |  |  |  |
| **Implementing yolo model for basic understanding of galaxy detection** | **6/12/22** | **20/12/22** |  |  |  |  |  |  |  |  |  |
| **Implement our own model architecture based on Neural network** | **21/12/22** | **27/12/22** |  |  |  |  |  |  |  |  |  |
| **Research on Squeezenet, Sparknet architecture and learnt about separable convolution** | **28/12/22** | **10/1/23** |  |  |  |  |  |  |  |  |  |
| **Worked on Squeezenet and mobilenet model for better accuracy** | **11/1/23** | **17/1/23** |  |  |  |  |  |  |  |  |  |
| **Final report preparation** | **18/1/23** | **6/2/23** |  |  |  |  |  |  |  |  |  |
|  | | | | | |  | | | | | |