Deep Convolutional Neural Networks for Galaxy Classification

***Abstract*—Deep convolutional neural networks (ConvNets) have proven to be effective in classifying stellar objects in binary classification problems. ConvNets can autonomously extract features from the data, reducing the need for human experts to be involved in classifying stellar objects. Using data from the Galaxy Zoo Data Release 2 and DESI Legacy Imaging Surveys (DECals), this paper proposes a model built upon previous models to produce accurate results in line with modern ConvNet performance on similar problems. Further research may bring the possibility of removing the human element entirely from classification of stellar objects.**

Keywords— Deep learning: Convolutional neural networks, Image convolution, Image Processing

I. Introduction

Traditionally classification of stellar objects has been done manually by humans. This requires trained personnel and time. Even with trained personnel and time, the vast quantities of data being collected by modern telescope arrays is staggering. Human classification of the data is too slow and costly.

Being able to automate all or part of the classification process frees up experts to work on other projects and allows for far more data to be processed. Neural networks also have the potential to outperform humans on classification tasks.

Deep convolutional neural networks (ConvNets) have demonstrated to be proficient in differentiating between different types of stellar objects. For example, a ConvNet can relatively easily distinguish between a galaxy and a star. However, distinguishing between types of galaxies can be more difficult. Many types of galaxies have traits that overlap. This make multi class classification significantly more difficult compared to binary classification.

Previous solutions have either been too narrow in scope, focusing on binary classification problems or have produced unsatisfactory results.

Data is sourced from the Galaxy10 DECals dataset. This dataset is a collection of images from three sources: Galaxy Zoo Data Release 2, DESI Legacy Imaging Surveys (DECals) campaign ab, and DECals campaign c. The data labels are crowdsourced by volunteers at Galaxy Zoo. The images are 256 x 256 pixels with red, green, and blue color channels. The galaxies are divided into 10 broad classes. Classes with small (<20) image counts were abandoned. The classes were tweaked to avoid sharing too many traits and overlapping with each other.

The dataset's images were center cropped to a more manageable size of 128 x 128 pixels. The training dataset was rotated 90 degrees to artificially increase the size of the training dataset.

The model is a ConvNet that convolves the input images to extract high level features which are then fed into the fully connected layer to learn the features.

The model was able to achieve an accuracy of ~75% and an F1-score of ~76%. Adding extra dropout layers, keeping the number of learnable parameters relatively low, and avoiding too many hidden layers kept the model from overfitting the training data.

The model was able to achieve a relatively favorable result. Future work may attempt to reduce the problem into smaller binary classification problems like many others have done solving similar problems.

II. State of Astronomical Image Classification

A. Available Datasets

Edwin Hubble's original classification of galaxies had them divided into broad categories [1]. The categories related to each other through time. With the relations taking on the shape of a hierarchical tree.

Bowen Goldade 40033178

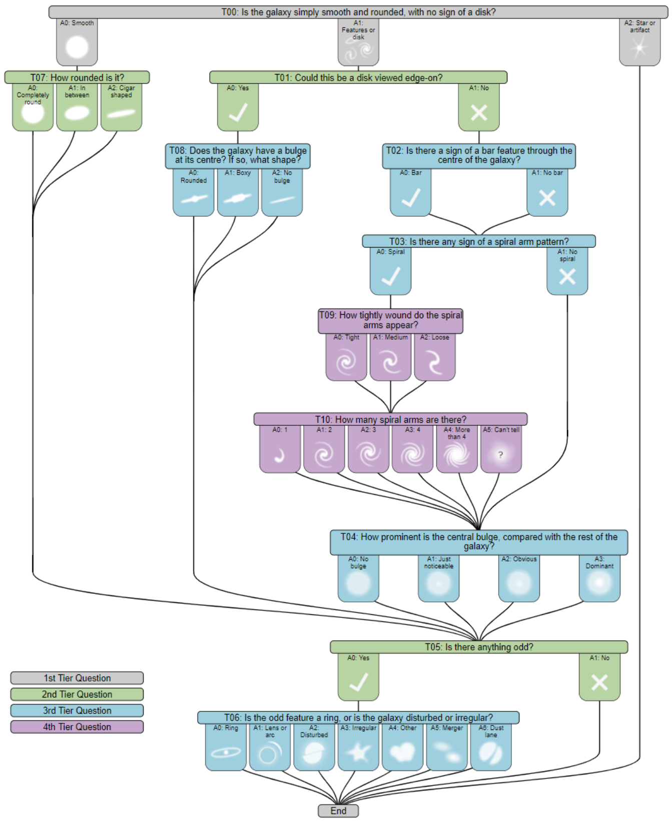


Fig. 1. Galaxy Zoo's classification decision tree.

Originally the classification was done by humans. With development of the internet and improvements in computing astronomers have been able to crowdsource classification problems. Galaxy Zoo is one such project that crowdsources classification of galaxies [2]. These systems use user feedback to build a probability model for classification. This model achieves comparable accuracy to those derived by expert astronomers [2].

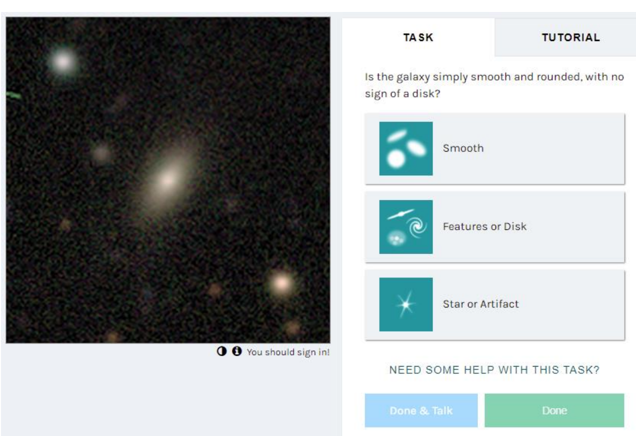


Fig. 2. Example of Galaxy Zoo's crowdsourcing tool.

The image cutouts used in Galaxy Zoo are collected from large telescope arrays. These arrays collect their data by scanning the night sky using multiple large telescopes. The resulting data are stored in vast databases cataloging the night sky. Telescope arrays like the Sloan Digital Sky Survey provide cutouts of individual objects for classification.

B. *Image Convolution*

Image convolution is the process of filtering an image in the spatial domain. A kernel is a matrix of size n x n. The kernel is passed over the image one pixel at a time. At each pixel, the kernel multiplies its values with the pixel value and its neighbors. Ultimately creating a new filtered image.

$$G_{Horizontal} = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} \quad (1)$$

$$G_{Vertical} = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (2)$$

Edge detection kernels (1) and (2) can be convolved with an image to create a new image which features the edges of the original. When combined with machine learning the values in a filter can be learned by the machine to extract desirable features.

C. *Deep Convolutional Neural Networks*

ConvNets have been used with increasing success for years now. They work by layering multiple image convolutions on top of each other. The output of one feeding into the next. Systems built in this way can autonomously and effectively extract features from an image that neural network can learn to classify the image.

The famous ImageNet challenge has contestants compete to classify more than one million high resolution images into 1000 classes. This has driven advancements with contestants proposing interesting solutions, like non-linear convolution models [3]. Other problems have been solved by deep convolutional neural networks with an accuracy higher than expert humans can achieve [4].

Naturally, the use of deep convolutional neural networks extends nicely to classifying images of stellar objects. The use of neural networks in combination with manual feature engineering is a long and difficult task. Kim & Brunner [5] are one of many to propose a solution using deep convolutional neural networks. They propose using deep convolutional layers to reduce manual feature engineering to a minimum. This allows them to categorize billions of stars and galaxies with minimal human involvement with accurate classifications.

III. Problem Motivation

As discussed above, we are collecting massive amounts of imaging data from the night sky. When classification of the data is done manually by experts a bottleneck is created. The time of these experts is scarce and valuable. Implementing a solution using ConvNets frees up their time to be better spent on other problems.

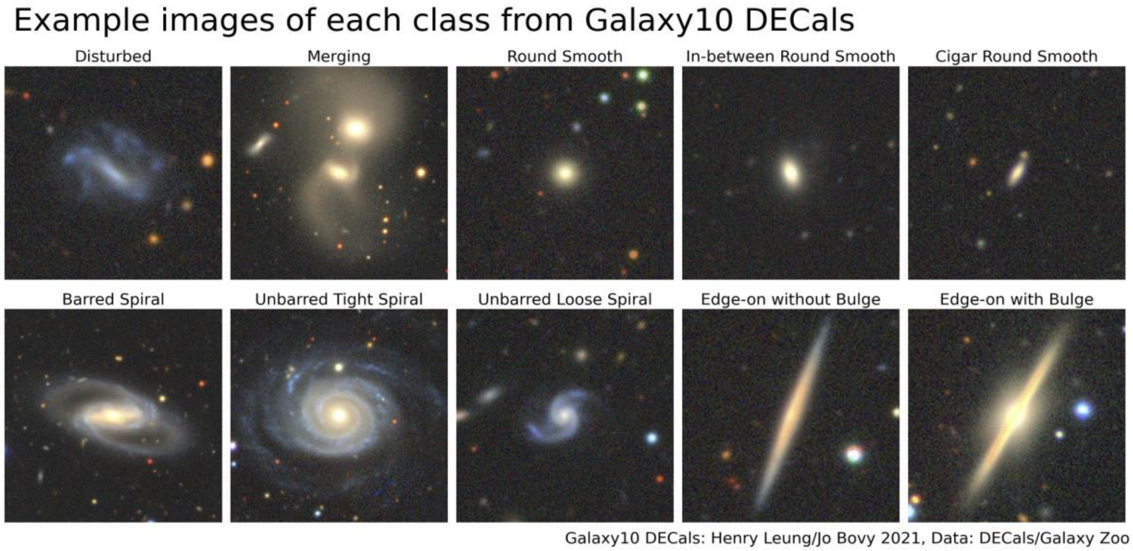


Fig. 3. Example of each Galaxy10 DECals class.

Unbarred Tight Spiral Galaxy Unbarred Loose Spiral Galaxy

Fig. 4. Example of types of galaxies difficult to distinguish.

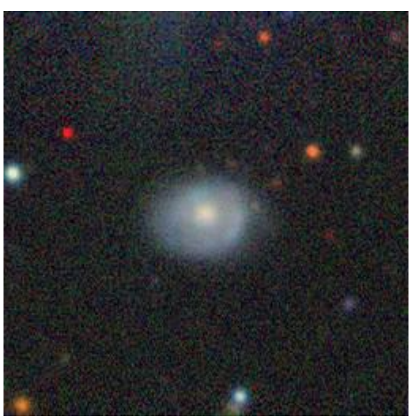


Fig. 5. Example of galaxy with common astronomical background noise.

The process of classifying stellar objects can be broken into smaller classification problems:

1. A location in the night sky is determined to have an object of interest or not. Creating a cut-out of the object to be passed on.
2. The cut-out is determined to be real and not an artefact or noise in the image.
3. The object in the cut-out is determined to be the type of object we are looking for. For example, a galaxy or a star.
4. The object's precise class is determined. For example, a barred spiral galaxy.

The first step can be accomplished with an algorithm locating light sources in the images. The second is a binary classification problem which can be solved with a similar application of ConvNets [6]. The third step is a multiclass classification problem. In this case the classes are relatively visually distinct from each other. Making classification relatively easier than the fourth and last step in the pipeline. In the last step the galaxies need to be distinguished from each other. This is a more difficult problem because many galaxy classes share traits with each other.

The cut-outs are relatively large images with 256 x 256 pixels with three color channels. At the input layer there are 196,608 data points for an algorithm to interpret. Using images of this size will take a significant amount of time to train an algorithm on.

The ConvNet may also be overfitting to other objects or noise in the background of the image. Fig. 5 demonstrates an example of an image with other objects in the cut-out and a normal amount of background noise. Therefore, reducing the complexity of the input images is important to the viability of a solution.

There are many efforts to apply ConvNets to automate the traditionally human task of classifying galaxies. I have chosen this last step which is the final classification of objects. In this project we have investigated applying ConvNets to solve this multi class classification problem.

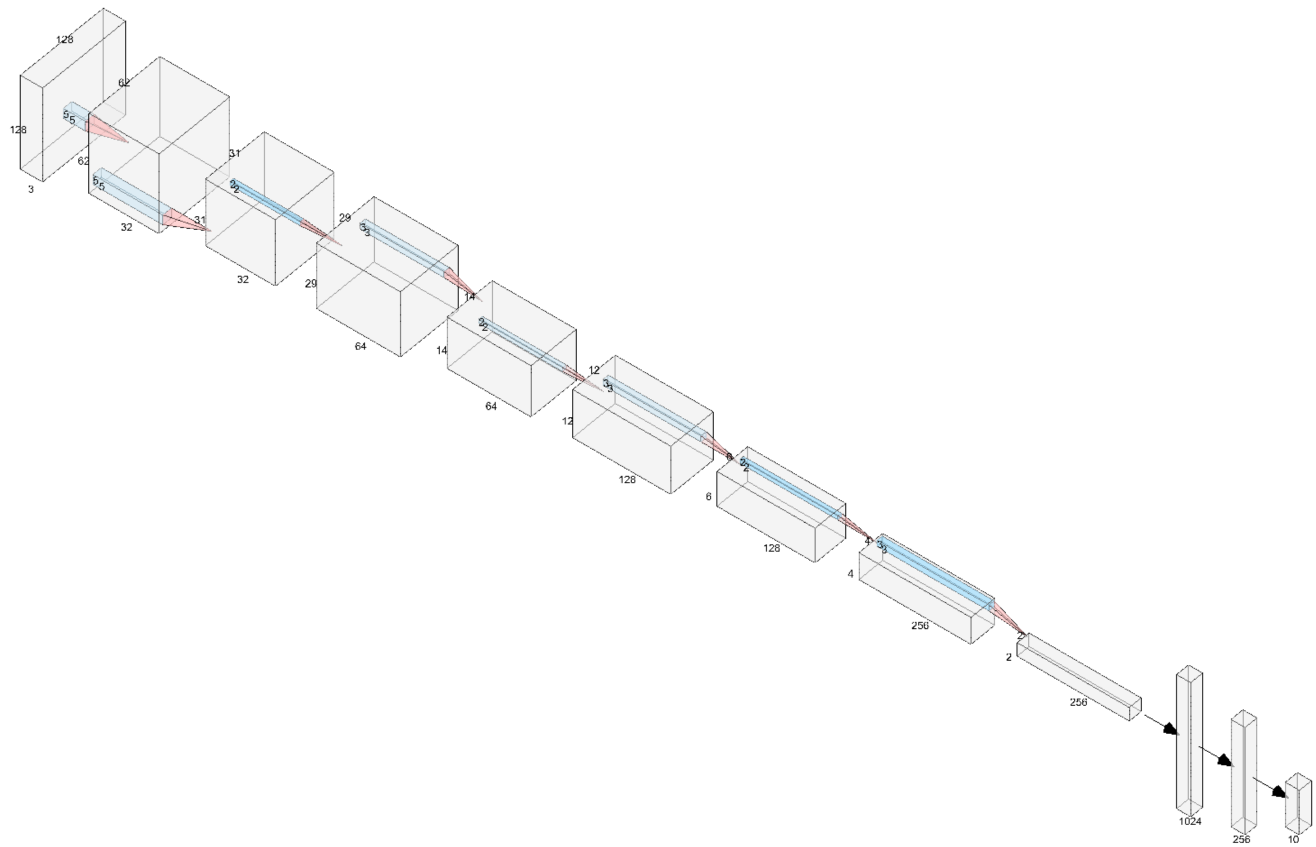


Fig. 6. Diagram of ConvNet architecture.

IV. Convolutional neural network framework

A. Dataset

The dataset selected is the Galaxy10 DECals dataset. This dataset is composed of 17,736 unique images of shape (256, 256, 3). The images are pulled from three sources: Galaxy Zoo Data Release 2, DESI Legacy Imaging Surveys (DECals) campaign ab, and DECals campaign c. The data labels come from Galaxy Zoo. The galaxies are divided into 10 broad classes demonstrated in Fig. 3. Classes with small (<20) image counts were abandoned. The classes were tweaked to minimize sharing too many traits and overlapping with each other.

B. Data Enhancement

The images as they come from the dataset are much larger than required for training a ConvNet. Reference [7] was able to achieve good accuracy in their model using input images of size (32, 32, 3). Achieved by cropping and zooming their input images of size (512, 512, 3).

The images were center cropped from (256, 256, 3) to (128, 128, 3) removing three quarters of the pixel values surrounding the galaxy. This resulted in a proportional improvement in training time for the model with a ~5% increase in accuracy.

To artificially increase the size of the training data, the training images were rotated 90/180/270 degrees. This is possible due to the rotation of a galaxy not changing its classification. This effectively increases the training dataset by a factor of four, creating new training images for the network to learn.

TABLE I. ConvNet Model Layer Summary

Layer	***Output Shape***	***# Param***
Convolutional	62 x 62 x 32	2432
Pooling	31 x 31 x 32	0
Convolutional	29 x 29 x 64	18496
Pooling	14 x 14 x 64	0
Convolutional	12 x 12 x 128	73856
Pooling	6 x 6 x 128	0
Convolutional	4 x 4 x 256	295168
Pooling	2 x 2 x 256	0
Flatten	1024	0
Dropout	1024	0
Fully Connected	256	262400

Layer	*Output Shape*	*# Param*
Dropout	256	0
Fully Connected	10	2570

The dataset was split with 80% in the training set, and 20% in the validation set. Image rotation was only applied on the training set.

C. Convolutional Neural Network Layout

The architecture of the model is presented in Fig. 6. The convolutional portion of the network is composed of four convolutional layers, each followed by a max pooling layer. The first convolutional layer has a kernel size of (5, 5) with a stride of 2. This first layer with its max pooling layer is used to rapidly decrease the complexity of the image. The image is rapidly decreased from (128, 128, 3) to (31, 31, 32). This allows the network to not linger on the fine detail of the image and quickly move on to higher dimensional features. Three more layers of Convolution, this time with a smaller kernel of (3, 3) and stride of 1, and max pooling reduce the image to a manageable 1024 data points.

After flattening the 1024 inputs are fully connected to a hidden layer with 256 nodes with a dropout with P = 0.1. This hidden layer is fully connected to the output layer with 10 nodes and a dropout with P = 0.25.

The activation functions used after each layer, with the exception of max pooling are rectified linear activation function (ReLU). ReLU sets negative values to zero while leaving positive values unmodified. This layer removes unwanted features from being passed down simplifying the calculations and improving training effectiveness.

For the final activation function, softmax was selected. Softmax is the standard activation function for multi class classification problems.

To guide the internal weights towards their ideal values, a loss and optimizer functions needed to be selected. A loss function of categorical cross entropy is standard for multi class classification. This function computes the difference between the model's predictions and the true label. The optimization function is used to adjust the model's internal weights to minimize the loss calculated by the loss function. Initially an optimizer of root mean square propagation (RmsProp) was used. This optimizer provided unsatisfactory convergence. The adaptive movement estimation (Adam) function with a learning rate of 0.001 was substituted and provided improved convergence.

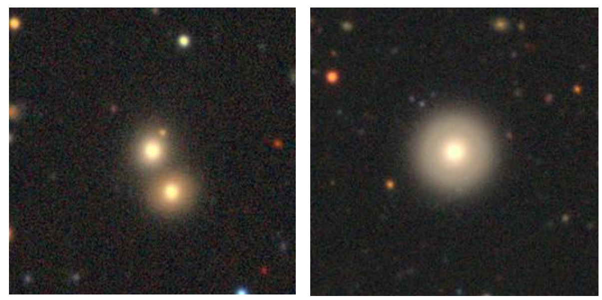
D. Avoiding Over/Underfitting

Overfitting occurs when the model has learned the training data too well. Resulting in great performance in the training data, but poor performance in the validation data. During testing it was found that having more hidden layers resulting in more parameters to learn caused overfitting in the model.

The overfitting was remedied by included a single hidden layer and introducing dropout. Dropout is a layer that with a set probability turns off a connection between two nodes. Increasing dropout forces the network to generalize its learning and avoid learning the data too specifically.

Underfitting occurs when the model does not have enough complexity, measured in learnable parameters, to learn the data effectively. This is remedied by increasing the size or number of hidden layers, or decreasing dropout, or by increasing the complexity of the image after the convolutional layers. During testing it was found that adding more than four convolutional and max pooling layers resulted in too few data points being output to the hidden layers.

V. EVALUATION



Merging Galaxy Round Smooth Galaxy

Fig. 7. Example of overlap involving merging galaxies.

Some categories of galaxies overlap with others. For example, merging galaxies are two or more galaxies in close proximity to each other. These two galaxies can take on the traits of other categories. This results in the model often incorrectly predicting that a merging galaxy is some other type of galaxy.

A. Evaluation Metrics

Accuracy is the easiest metric to evaluate and understand. It is the number of correct predictions divided by the number of total predictions.

$$Accuracy = \frac{True\ Positives + True\ Negatives}{Total\ Predictions} \quad (3)$$

The model was able to achieve an accuracy of ~75%. This is a relatively good score. This demonstrates the model achieved relative success in classifying the images.

Precision measures how many positive predictions were correct. This metric conveys how many true positives to false positives the prediction generates.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \quad (4)$$

The model achieved a precision of ~85%. This means when the model predicts the class of a galaxy it is the correct class ~85% of the time.

Recall measures the opposite of precision. It measures the proportion of true positives to false negatives, or how many galaxies of a given class were correctly classified.

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \quad (5)$$

The model was able to achieve a recall of ~68%. Noticeably less than the achieved precision. This is due to precision and recall being inversely related. The fewer false positives the model predicts, the better the precision. Decreasing false positives has a knock-on effect of increasing false negatives which reduces recall. The model over emphasizes minimizing false positives. Minimizing false positives will keep the data being placed in a class as correct as possible. This allows the resulting labels to be used to further train more models. While false negatives do not interfere as significantly with any training being done on the resulting labels.

Combining both precision and recall into a single metric creates the F1-score. F1-score is the harmonic mean of precision and recall.

$$F1 - score = 2 * \frac{precision * recall}{precision + recall} \quad (6)$$

The model achieved an F1-score of ~76%. The F1-score demonstrates a good overall result for precision and recall.

B. Results

Accuracy was the metric used to evaluate performance of the model during training. After 50 epochs the loss and accuracy had converged. Training was stopped here to avoid overfitting the training data.

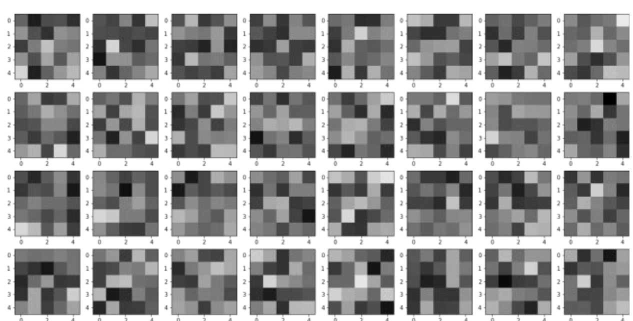


Fig. 8. The 32 filters with kernel 5 x 5 learned by the model.

Peering under the hood of the model for a moment. We can take a look at the filters it is using to extract the lowest level features.

VI. Related Work

Using ConvNets to classify images is a very broad topic of study. Reference [3] has proposed an extremely generalized model for the ImageNet competition. Their model uses multiple parallel convolution and max pooling layers. This allows their model to extract difference sized features simultaneously. The problem they are solving is more generalized and requires their model to learn significantly more parameters. Their model has 60 million learnable parameters, while this model has 650 thousand. This reflects the difference is scope of the problems being solved.

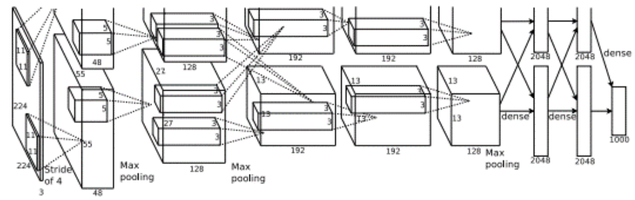


Fig. 9. Reference [3] ConvNet architecture for ImageNet challenge.

Reference [6] proposes using ConvNets to clean the data before being passed to other models for classification. Their model able to distinguish between real stellar objects and noise or errors in the image. These errors can be caused by residual light from a star as the telescope moves. With their model they achieved an accuracy of 92%. This model notably has far fewer nodes in its hidden layer than ConvNets designed for similar purposes. Which may indicate determining if an object is noise may be a relatively easier task for a neural network to learn.

TABLE II. NEGM LAYER FRAMEWORK

Layer	*Output Shape*	*# Param*
Convolutional	69 x 69 x 32	896
Pooling	35 x 35 x 32	0
Dropout	35 x 35 x 32	0
Convolutional	35 x 35 x 64	18496
Pooling	18 x 18 x 64	0
Dropout	18 x 18 x 64	0
Convolutional	18 x 18 x 128	73856
Pooling	9 x 9 x 128	0
Dropout	9 x 9 x 128	0
Flatten	10368	0
Fully Connected	128	1327232
Dropout	128	0
Fully Connected	2	258

Reference [5] proposes using ConvNets to avoid manual feature engineering. They identify the problem of manual feature engineering requiring careful engineering and expertise to transform images into suitable features to be used in neural networks. Their problem differs slightly in that they are classifying source images which include both stars and galaxies, rather than types of galaxies. Their proposed ConvNet involves many more convolutional layers in-between max pooling layers. This allows their model to learn more filters while extracting features. Their model ultimately classifies objects into star or galaxy classes. The objects it classes as galaxies could easily be fed as input into this project's model, to further classify the galaxy type.

TABLE III. KIM & BRUNNER LAYER FRAMEWORK

Layer	***Filters***	***Kernel***	***Padding***
Convolutional	32	5 x 5	-
Convolutional	32	3 x 3	1
Pooling	-	2 x 2	-
Convolutional	64	3 x 3	1
Convolutional	64	3 x 3	1
Convolutional	64	3 x 3	1
Pooling	-	2 x 2	-
Convolutional	128	3 x 3	1
Convolutional	128	3 x 3	1
Convolutional	128	3 x 3	1
Pooling	-	2 x 2	-
Fully Connected	2048	-	-
Fully Connected	2048	-	-
Fully Connected	2	-	-

Reference [7] attempts to solve the same problem presented in this work. Namely, how do you accurately classify many different types of galaxies into many similar classes. They took a similar approach to other by breaking the problem down into a binary classification problem. Ultimately, they achieved an accuracy of 98% with a ConvNet. Their 37 initial classes were reduced to galaxies being smooth or galaxies having features or a disk. The model they propose uses three convolutional layers each followed by a max pooling layer with only a single fully connected layer.

VII. FUTURE WORK

State your ideas of what you estimate could be done in the future in order to further improve on the addressed problem. You may also state further problem areas you identified to be researched in the future. This section is not mandatory but may be useful for others in identifying interesting new research problems.

A. *Denoising Input Images*

By incorporating denoising strategies on the input data it is possible to artificially increase the exposure time of the images. Powerful telescopes are in short supply. Reference [8] proposes a model to reduce image noise to a level comparable to doubling the exposure time.

Increasing the signal to noise ratio in the input images allows for improvements to model performance. With less noise in the input images the model may have fewer features to learn. The model learning to fit noise becomes less of a problem allowing for greater accuracy of classification.

B. *Layering Convolutional Neural Networks*

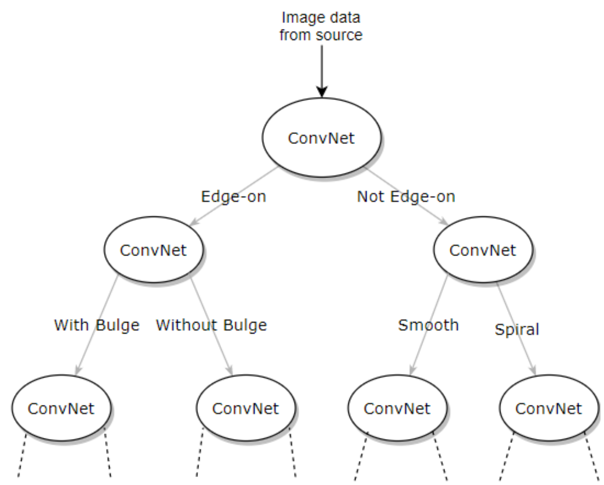


Fig. 10. Reference [3] ConvNet architecture for ImageNet challenge.

Many of the authors examined here reduce the problem of classification into a binary classification problem. These binary classification models are achieving significantly improved performance.

If reducing the problem to binary classification gives excellent performance results, and the process of galaxy classification can be approximated as a hierarchical tree. I would propose a model of hierarchically layered ConvNets. Each ConvNet would split the dataset into as few classes as possible.

While training a ConvNet to classify many types of galaxies is achievable, the resulting model will be large and complicated. This new large model will be difficult to train and tune. Using multiple smaller models would allow for easier training and tuning of hyperparameters.

VIII. CONCLUSION

Modern telescope arrays are collecting truly massive amounts of data. Without applying autonomous feature extraction and machine learning algorithms we would not be able to keep up with the data. The ease of scanning the night sky in the visible light spectrum creates an abundance of data which can be used to train machine learning algorithms. Moving into the future, classification of the vast majority of stellar objects will need to be done by machines rather than humans.

There currently exists systems which can accurately pick out stellar objects out of a scan of the night sky. Classifying these objects is still a difficult problem where humans perform better than machine learning algorithms.

Attempts are being made to reduce this workload by employing deep convolutional networks to autonomously extract features from cut-outs of these sky surveys. By training a neural network to convolve these images and to learn which filters should be used to achieve the desired results autonomous classification can be achieved.

The architectures of proposed solutions are similar in many ways. Many solutions simplify the problem of classifying everything in the night sky down to a binary classification problem. The ConvNets, when solving this size of problem do not need to learn as many features to be able to distinguish between classes. This allows them to achieve better performance results.

I propose a solution to the multi class classification problem of classifying 10 of the more common types of galaxies. Using four convolution layers the model is able learn high level features of the galaxies. These features are then able to be used by the neural network to classify the galaxies relatively accurately.

The relatively small training dataset is a potential issue. The training dataset is on the order of 10^4 while the data that has been labeled solely by galaxy Zoo is at least on the order of 10^5.

The model was able to achieve a relatively high accuracy of 75% and F1-score of 76%. The model overemphasized minimizing false positives by achieving a precision of 85% and underemphasized false negatives by achieving a recall of 68%.

Future work can explore the potential benefits of breaking the multi class classification problem up into more manageable binary classification problems. This could potentially speed up training and tuning of the model and allow for higher prediction accuracy.

References

[1] E. Hubble, "Extragalactic nebulae.", The Astrophysical Journal, vol. 64, p. 321, 1926. Available: 10.1086/143018. J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[2] C. Lintott et al., "Galaxy Zoo: morphologies derived from visual inspection of galaxies from the Sloan Digital Sky Survey", Monthly Notices of the Royal Astronomical Society, vol. 389, no. 3, pp. 1179-1189, 2008. Available: 10.1111/j.1365-2966.2008.13689.x.

[3] A. Krizhevsky, I. Sutskever and G. Hinton, "ImageNet classification with deep convolutional neural networks", Communications of the ACM, vol. 60, no. 6, pp. 84-90, 2017. Available: 10.1145/3065386.

[4] A. Buetti-Dinh et al., "Deep neural networks outperform human expert's capacity in characterizing bioleaching bacterial biofilm composition", Biotechnology Reports, vol. 22, p. e00321, 2019. Available: 10.1016/j.btre.2019.e00321.

[5] E. Kim and R. Brunner, "Star–galaxy classification using deep convolutional neural networks", Monthly Notices of the Royal Astronomical Society, vol. 464, no. 4, pp. 4463-4475, 2016. Available: 10.1093/mnras/stw2672

[6] Y. Negm, "Astronomical Images Classification", Medium, 2021. [Online]. Available: https://medium.com/analytics-vidhya/astronomical-images-classification-18d39a7f323f. [Accessed: 25- Apr- 2021]

[7] H. Shi, "Galaxy classification with deep convolutional neural networks," M.S. thesis, Graduate College, Univ. Illinois, Urbana-Champaign, Mar. 2016. Accessed on: Apr. 25, 2021. [Online]. Available: https://core.ac.uk/download/pdf/158314025.pdf

[8] A. Vojtekova et al., "Learning to denoise astronomical images with U-nets", Monthly Notices of the Royal Astronomical Society, vol. 503, no. 3, pp. 3204-3215, 2020. Available: 10.1093/mnras/staa3

[9] A. Vojtekova et al., "Learning to denoise a

[10] G. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors", CoRR, vol. 12070580, 2012. Available: http://arxiv.org/abs/1207.0580. [Accessed 26 April 2021].

[11] C. Lintott et al., "Galaxy Zoo 1: data release of morphological classifications for nearly 900 000 galaxies", Monthly Notices of the Royal Astronomical Society, vol. 410, no. 1, pp. 166-178, 2010. Available: 10.1111/j.1365-2966.2010.17432.x [Accessed 26 April 2021].

[12] Walmsley, M., “Galaxy Zoo DECaLS: Detailed Visual Morphology Measurements from Volunteers and Deep Learning for 314,000 Galaxies”, *arXiv e-prints*, 2021

[13] Dey, A., “Overview of the DESI Legacy Imaging Surveys”, *The Astronomical Journal*, vol. 157, no. 5, 2019. doi:10.3847/1538-3881/ab089d.