Rice Classification

Blaise Duncan, Marc Rodriguez, James Stockwell
SDSU CS 549

Abstract - Rice is a dish that is popular and essential to many cultures throughout the globe. It is a dish that has been around for thousands upon thousands of years across many cultures and generations. With time, it is crucial to note that there have been variances in the different types of rice that have been developed. Through the course of this project, we aim to create software that can reliably differentiate the various types of rice. While there are thousands of varieties of rice, there are a handful of common ones used for different culinary applications, though laymen could be forgiven for being able to tell them apart. With the rising power of image recognition machine learning algorithms in recent years, it stands to reason that small, simple images that could be categorized efficiently, such as those of rice, could be digitally processed for quick and efficient sorting. This is done through the implementation of CNN or, in other words, a convolutional neural network in which raw input images are taken as a means to derive a classification. Assuming the functionality of the model, we will be testing it by viewing its ability to classify the different variations of rice given the parameters it has been allotted. This will be done through a series of various techniques in which the power of machine learning will be exploited. In testing this hypothesis or problem statement, we have begun by testing various types of models and training them with a series of 75,000 images of 5 different variations of rice.

I. Introduction

As stated previously, rice is a dish that has been around for generations and has played a crucial role in becoming a staple in various cuisines worldwide. It is to be reiterated that with time, various variations of rice have become prominent in our world, and each comes with different features to the next. In the past, rice would often have to be classified through manual inspection, which would be heinously time-consuming in the sense that rice is extremely tiny in its individual forms. Measuring and classifying each grain accurately would likely require an immense amount of time and costs in general labor, not accounting for the tribulations of human error. However, through the development of machine learning and the incorporation of various techniques, we can now develop models that accurately perform rice classification using the variables attributed to individual features of the rice.

Our current goal is to generate the facilitation of a model that would be able to accurately account for the different variations of rice through classification. We are aiming to do this by implementing the techniques of SVM and KNN, which can be denoted as Support Vector Machines and K-nearest neighbors; extraordinarily popular subjects in the realm of machine learning and its affiliated concerns. In their associated context, SVM can be defined as a classification algorithm that can also be utilized in regression. SVMs tend to utilize what is known as a hyperplane to separate certain classes, depending on the classifiers and the variable parameters. As for KNNs, they are utilized for both classification and regression as well. In their individual techniques, they gather the K, which can be denoted as the nearest neighbors of data points to calculate the mean. In other words, it can be described as being in charge of storing all training data points and eventually computing distances between new data points and training data points.

II. Task Descriptions of the Techniques Implemented

SVM:

- Through the implementation of SVM, we will be utilizing a series of multi-class classifications, given that we are working with a series of various types of rice. These will be attributed to Arborio, Basmati, Ipsala, Jasmine, and Karacadag types of rice. They are to be categorized and placed into their own associated classes depending on their features.

CNN

- By training our model with an immense amount of images, the Convolutional Neural Networks will be able to classify the images of the different types of rice. Again, each respective image will be able to fall under one of the five categories of Arborio, Basmati, Ipsala, Jasmine, and Karacadag types of rice.

KNN

- The goal is to develop an identification system through the use of a K-Nearest Neighbor implementation through the dataset. Here the task could be described as extracting the most prominent features from each of the labeled images to utilize in the classification algorithm. This process is to be repeated and eventually evaluated based on the accuracy.

A. General Observations Attributed to Rice Classification

An observation can generally be defined as a generalization of an instance of data. To further test our understanding and implement these variables into our model, we will begin observing features of the common rice used within our model. Below is a table that illustrates the different types of rice that will be used within the data set. The description of the individual features of each variant is to be noted, given that it will be crucial to the observations and classifications done by the model.

Arborio	Basmati	Ipsala	Jasmine	Karacadaq
A short grain of rice with a somewhat rounded shape. Notably is starker white in the middle and translucent at the edges.	Longer and thinner than the arborio rice with slightly pointed tips. Uniformly white throughout the grain.	Short and similar in shape to the arborio though noticeably larger. Is uniformly white like the basmati.	Another long grain rice like the basmati, but is slightly more round and oval shaped. May also be slightly darker than basmati on average.	The shortest grain of rice in the dataset and the most round. Has a similar opaqueness gradient to arborio, but not as consistent or noticeable.

B. Notable Equations

SVM

$$J(w, b, \alpha) = \frac{1}{2} w^T w - \sum_{i=1}^{N} \alpha_i d_i (w^T x_i + b) + \sum_{i=1}^{N} \alpha_i$$

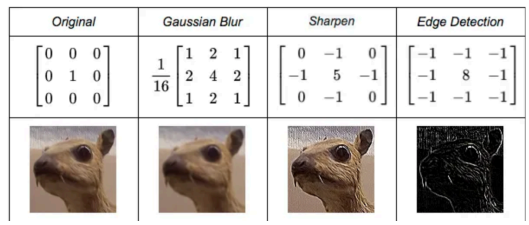
$$\text{At the optimum } \frac{\partial J}{\partial w} = 0 \text{ and } \frac{\partial J}{\partial b} = 0$$

$$\Rightarrow w_o = \sum_{i=1}^{N} \alpha_i d_i x_i \text{ and } \sum_{i=1}^{N} \alpha_i d_i = 0$$

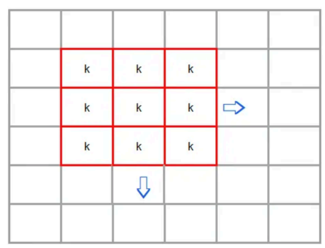
KNN

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

CNN



Examples of filters (For our task, some convolutional layers have predetermined values, and others have trainable values).



How a filter moves across an input image. The feature map is derived from the dot product between the overlapping region of the input image and the kernel's values.

III. Challenges

It is crucial to note that while models work extraordinarily well in the realm of classification, just like humans, they are prone not to be perfect and eventually make a few errors. Regarding likely challenges that may occur, we must begin by understanding that some of the grains tend to be similar in their features and individual sizes. This can be depicted in basmati and jasmine rice as an example. Arborio and Ipsala are other likely areas of confusion. With this being stated, it is crucial to go through each of the differing algorithms and explain where areas of challenge or general confusion can occur.

Challenges can be further depicted in the SVM in the sense that a common downside of this mode seems to be in relation to general model complexity and scalability. We are only working with five sets of variations of rice, but given an immense amount of data, an SVM may be prone to falling short of being able to accurately represent the large-scale training given the fact that memory is a crucial issue to consider. Hyperparameters of the model may also be a sensitive issue in the sense of tuning. However, given the similarity of these issues, we could likely prevent them by simplifying the model and general data set.

Challenges to be noted with the aspect of KNN could be attributed to the general scalability given that a larger dataset may result in a larger cost to compute the overall solution of the algorithm. Classes may also face a set of challenges associated with a bias that may occur if certain images of rice are not classified correctly. This would likely be tied to the presence of any falsely classified grains of rice given their size or general features. Suitable characteristics for the rice must be employed to reduce this issue through proper implementation within the data.

Lastly, a CNN may face difficulties with overfitting, especially if the dataset is limited, as the CNN may begin to fit the noise in the training dataset and fail to generalize to new data.

A. Solutions to Address Challenges

Overall it can be concurred that with the usage of these different types of models, implementations can be foreseen in addressing changes to the parameters of the given models. Speculations of possible solutions would likely be attributed to changing or altering the sizes of the given data set or model given that bias or variance can occur. However it is crucial to note that the majority of the solutions could be attributed towards fine tuning parameters based upon where one is content with the allotted accuracy of the model and its concerns. Combinations of various KNN model predictions can be done as a means to get a basic understanding of where the predictions are to be averaged if fluctuation is to occur in any given results. Regularization and general data augmentation could be implemented into any of the techniques involving our CNN to better generalize the data.

IV. Experiments

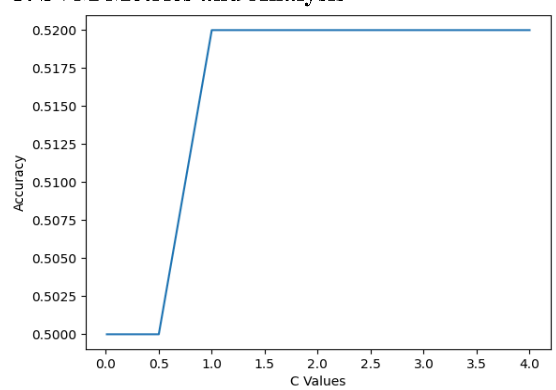
A. Dataset Description

The rice dataset we're using contains a total of 75,000 images of single grains of rice. These images are split into 5 groups of 15,000 images each representing a different variety of rice. The varieties in question are arborio, basmati, ipsala, jasmine, and karacadag [1].

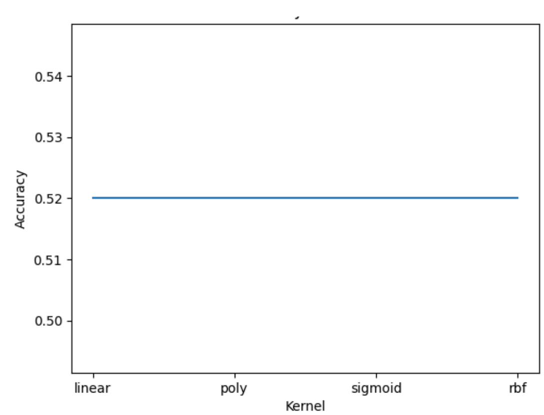
B. SVM Model Description

Support vector machine. A model that attempts to establish a dividing hyperplane between binary data. Since our data is multiclass, we went with an OvO-style model, which makes an SVM model for each class and combines the results of each to form a classification. The hyperplane can take on many broad shapes depending on the type of kernel used, while the C value determines how much tolerance the model will have for noise. A test with a subset of the data was used to determine the best kernel and C value for our model.

C. SVM Metrics and Analysis



C Value Test Results



Kernel Value Test Results

Higher C values (at least those less than 4) give more accurate results. Considering that there is a lot of variance and similarities between the classes, this is unsurprising. What is a surprise however is that the type of kernel seems to have no effect on the accuracy of this model.

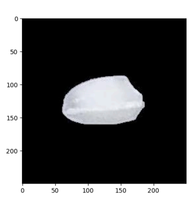
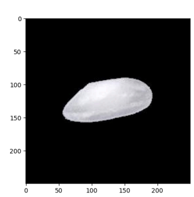
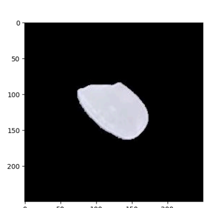
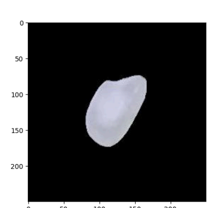
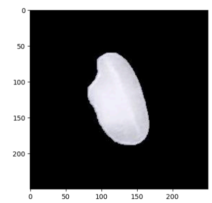
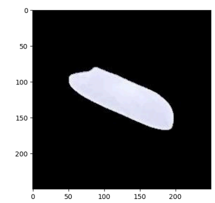
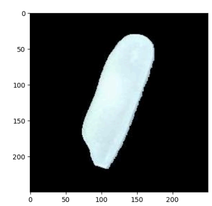
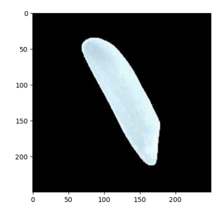
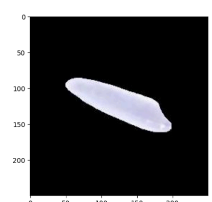
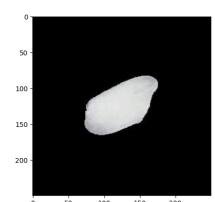
Confusion Matrix

	Predicted Class				
True Class	Arborio	Basmati	Ipsala	Jasmine	Karacadag
Arborio	2874	3	0	0	118
Basmati	1	2633	124	225	18
Ipsala	0	10	2940	1	16
Jasmine	0	76	0	2942	2
Karacadag	223	119	4	0	2660

Accuracy =.9367

Precision and Recall

	Arborio	Basmati	Ipsala	Jasmine	Karacada g
Recall	.9596	.8774	.9909	.9742	.8820
Precision	.9247	.9268	.9583	.9287	.9453

	
True: arborio	True: arborio
Predicted: karacadag	Predicted: jasmine
	
True: arborio	True: karacadag
Predicted: karacadag	Predicted: arborio
	
True: arborio	True: arborio
Predicted: karacadag	Predicted: karacadag
	
True: jasmine	True: jasmine
Predicted: basmati	Predicted: basmati
	
True: basmati	True: basmati
Predicted: jasmine	Predicted: jasmine

Error Sample Images

D. SVM Thoughts

SVM was shown to be surprisingly accurate in this visual classification scenario. While we had doubts about the model's ability to draw clear hyperplanes in samples with so much muddy variance, it is clear that it is able to do so without much tweaking. That being said, it does noticeably stumble when comparing grains of similar size. Understandably so, given that size and shape are the main two factors a human would need to go off of, but imperfect nonetheless.

E. CNN Model Description

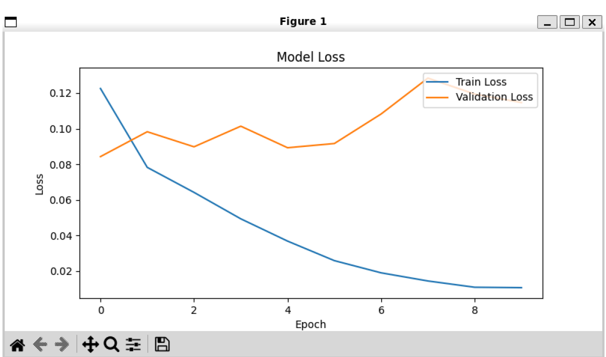
A convolutional neural network is a neural network that contains convolutional layers. These layers use a filter that "slides" across the input data to generate a feature map. We used Keras's Sequential class to implement the model and define our architecture. The network architecture used in our experiment consisted of a combination of convolutional layers with relu activation functions and max-pooling layers, which go into a dense layer with 128 nodes after flattening, which finally leads into the output layer with 5 neurons using a softmax activation function. We used Kera's utility function image_dataset_from_directory(), to feed the data into the model using a TensorFlow Dataset object. This allowed for easy management of testing and training splits and feeding the images in batches.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 254, 254, 16)	448
max_pooling2d (MaxPooling2D)	(None, 127, 127, 16)	0
conv2d_1 (Conv2D)	(None, 125, 125, 32)	4,640
max_pooling2d_1 (MaxPooling2D)	(None, 62, 62, 32)	0
conv2d_2 (Conv2D)	(None, 60, 60, 16)	4,624
flatten (Flatten)	(None, 57600)	0
dense (Dense)	(None, 128)	7,372,928
dense_1 (Dense)	(None, 5)	645

Total params: 7,383,285 (28.16 MB)
Trainable params: 7,383,285 (28.16 MB)
Non-trainable params: 0 (0.00 B)

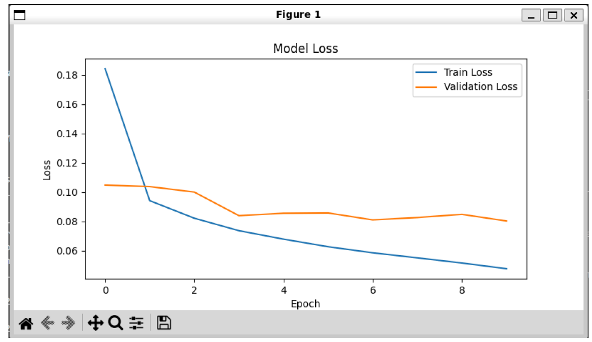
F. CNN Metrics and Analysis

The initial model architecture has 256 nodes in the second-to-last dense layer, effectively doubling the trainable parameters. When fitting the model, there was a very high level of overfitting occurring, with training accuracy as high as 99%, but with the validation error increasing with each epoch.



Loss over epochs; first attempt

To remedy this, I employed two strategies to reduce the model's complexity. First, I reduced the number of nodes in the second-to-last layer to the current 128. Secondly, I introduced l2 regularization to the convolutional layers. This resulted in less overfitting, and after increasing the l2 parameter to 0.2 from 0.1, the validation loss was acceptable, albeit it could be reduced further, and further experimentation could include adding an additional max pooling layer after the 3rd convolutional layer.



Loss over epochs; after adjustments

Confusion Matrix

	Predicted Class				
True Class	Arborio	Basmati	Ipsala	Jasmine	Karacadag
Arborio	1503	0	1	6	37
Basmati	1	1552	0	20	0
Ipsala	4	0	1465	6	0
Jasmine	6	45	3	1410	0
Karacadag	57	0	0	0	1372

Accuracy = .9752

Precision and Recall

	Arborio	Basmati	Ipsala	Jasmine	Karacada g
Recall	.9716	.9866	.9932	.9631	.9601
Precision	.9567	.9718	.9973	.9778	.9737

	
True: arborio	True: karacadag
Predicted: karacadag	Predicted: arborio
	
True: karacadag	True: arborio
Predicted: arborio	Predicted: karacadag
	
True: jasmine	
Predicted: ipsala	

Error Sample Images

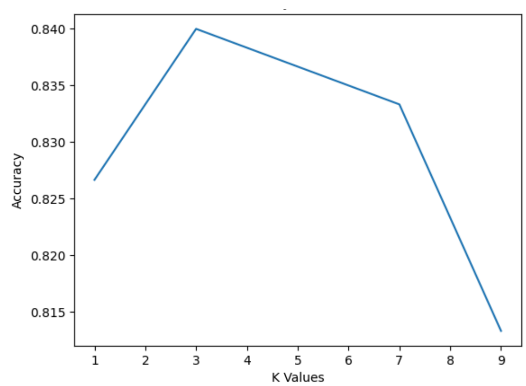
G. CNN Thoughts

Overall, CNN performed the best on the dataset compared to the other two methods. This outcome was expected, given the complexity of the model compared to the other methods. The training time, especially for those without access to GPUs, could be a factor that dissuades one from utilizing CNN. However, the difference in the resulting accuracy arguably justifies using this method over others.

H. KNN Description

K nearest neighbors is a simpler model that determines a new node's class via the classes of its nearest neighbors. The parameter k determines how many closest neighbors the model will consider in this process. We train multiple models from a small subset of the data to determine the ideal k value to use in the final model.

I. KNN Metrics and Analysis



K Value Test Results

A K value of 3 seems to be most ideal for the purposes of this model.

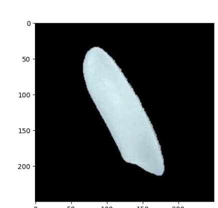
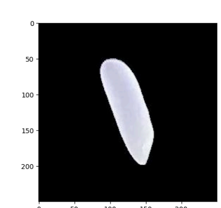
Confusion Matrix

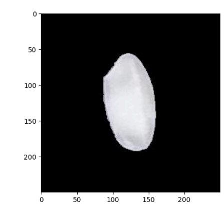
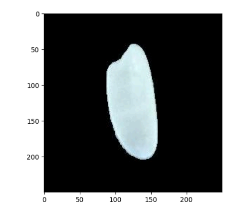
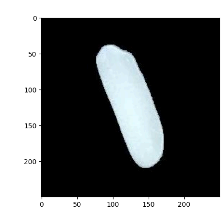
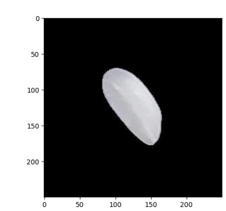
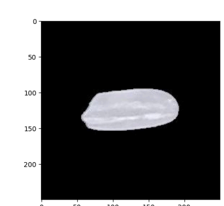
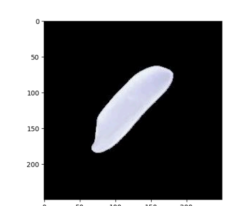
	Predicted Class				
True Class	Arborio	Basmati	Ipsala	Jasmine	Karacadag
Arborio	2813	12	21	155	1
Basmati	18	2949	19	2	0
Ipsala	30	3	2866	0	177
Jasmine	80	0	0	2856	0
Karacadag	0	0	106	0	2891

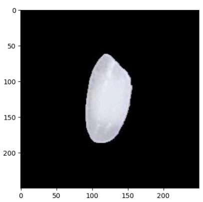
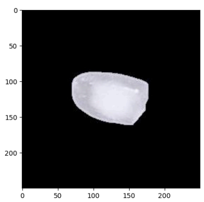
Accuracy = .9584

Precision and Recall

	Arborio	Basmati	Ipsala	Jasmine	Karacadag
Recall	.9370	.9869	.9317	.9728	.9646
Precision	.9565	.9949	.9515	.9479	.9420

True: jasmine	True: basmati
Predicted: basmati	Predicted: jasmine
	
True: arborio	True: jasmine
Predicted: karacadag	Predicted: arborio
	
True: jasmine	True: arborio
Predicted: basmati	Predicted: karacadag
	
True: arborio	True: basmati
Predicted: jasmine	Predicted: jasmine

	
True: arborio	True: arborio
Predicted: karacadag	Predicted: karacadag

Error Sample Images

J. KNN Thoughts

Based upon the values that have been outputted by the given equations and the performance of the model, the KNN method also tends to hold merit in establishing a high amount of accuracy. While in a negative contrast, the KNN model would be susceptible to the given hindrances of bias or other factors that could be attributed to a large data set or implemented parameters, our model does a rather decent job of classifying the rice. Overall, the implementation of the KNN would be a good approach to deciphering the solution of rice classification through machine learning and its associated techniques.

V. Conclusion

Through the means of this report we established a way to classify the variations found in popular rice grains across the globe. With rice being one of the most consumed dishes across the planet, it was crucial to develop a system that could accurately derive and classify the different variations of rice. In this report, we were able to implement a series of popular models in the realm of machine learning to classify the rice variations with high accuracy. The implementation of models such as the SVM, KNN, and CNN models and their affiliated techniques paved the way to generate a methodology that would accurately identify the different subsets of rice and classify them to their best extent. In this report, representations of the particular data and resulting calculations can be depicted in relation to the corresponding technique of implementation. This project was just one of the many methods that could be used to further evaluate the usage of such techniques and their regards.

VI. Team Contributions

Blaise Duncan- CNN model creator and team planner
Marc Rodriguez - KNN model creator and report writer
James Stockwell - SVM model creator and report formatter

VII. References

[1] M. Koklu, I. Cinar, Y.S Taspinar, “Rice Image Dataset.” Distributed by Kaggle. https://www.kaggle.com/datasets/muratkokludataset/rice-image-dataset/data