**National College of Ireland**

**MSc Project Submission Sheet**

**School of Computing**

**Student Name:** Tom MacNamara

**Student ID:** x19144768

**Programme:** MSCDATOP **Year:** 2022

**Module:** MSC Research Project

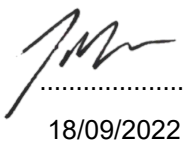
**Supervisor:** Dr Catherine Mulwa

**Submission Due Date:** 19/09/2022

**Project Title:** An Assessment of Classification Approaches in Identifying Martian Geological Features

**Word Count:** 4562 **Page Count** 29

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

**Signature:** 

**Date:** 18/09/2022

**PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST**

| | |
|---|---|
| Attach a completed copy of this sheet to each project (including multiple copies) | □ |
| **Attach a Moodle submission receipt of the online project submission,** to each project (including multiple copies). | □ |
| **You must ensure that you retain a HARD COPY of the project**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. | □ |

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

| **Office Use Only** | |
|---|---|
| Signature: | |
| Date: | |
| Penalty Applied (if applicable): | |

# An Assessment of Classification Approaches in Identifying Martian Geological Features

Tom MacNamara - x19144768

MSCDATOP - Semester 2 2022

**Abstract**

Image classification is a field with many viable methods towards solving problems within it. It is therefore important to analyse and determine whether there exists a best machine learning technique which can be applied when categorising images. This report analyses nine approaches: K-nearest Neighbours; Support Vector Machine; Random Forest; Logistic Regression; Gaussian Naive Bayes; Multinomial Naive Bayes; Complement Naive Bayes; LightGBM; and a Convolutional Neural Network to determine if one of these approaches is ideal in classifying images of the surface of Mars.

## 1 Introduction

This project aims to analyse multiple image classification processes to determine which is the most viable in classifying images of the Martian surface as found in datasets published by NASA. Image classification is a subsection of computer vision, whereby images are analysed to determine whether specific features are present in those images. Classification specifically refers to images that have known potential contents, i.e., a dataset

will contain images of a city and images of a rural area. It is known that the image will be sortable into one of two or more buckets (ScienceDirect 2022). This is different to object detection, where while the goal is still to identify the semantic contents of an image, a bounding box is drawn around the object rather than the images themselves being categorised (Papers With Code 2022). This does not necessarily address the same problem. Classification algorithms may examine a photograph of a dog and identify the breed. Detection algorithms may examine a photo and identify whether a dog is present.

Image classification is a complex problem with many possible different approaches to solving. Some approaches feature complex techniques such as Neural Networks (S. Xie and Tu 2015); some utilise simpler techniques, such as Support Vector Machine (SVM) or Random Forest (RF) (Mercier and Lennon 2003). While there are many studies analysing the efficacy of these models, there is a knowledge gap with regards to direct comparisons. Image classification studies typically feature one technique carried out on one dataset. Like for like comparisons using the same metrics using different models, each trained on the same data, are absent from the body of knowledge.

## 1.1 Research Question

Classifying images is a problem that appears in a wide array of fields. In recent years there has been an increase in classification problems thanks to the explosion in popularity of the Internet Of Things (IOT). Identifying which frames of a home security video feature an intruder; whether an obstruction is blocking an autonomous vehicle; whether cancer is visible in an x-ray. These are all examples of commercial applications of image classification where a clear direction on model choice would be a valuable boon. This presents the question:

**RQ:** *"To what extent can direct comparisons between modelling techniques identify a clearly favourable method when applied to an image classifica-*

*tion problem?"*

## 1.2 Research Objectives

1. Conduct an investigation and state of the art review to determine the landscape of the field.
2. Pre-processing of images to allow input into a model.
3. Implementation and evaluation of modelling approaches.
   (a) Implementation and evaluation of a Convolutional Neural Network (CNN) model.
   (b) Implementation and evaluation of a K-Nearest Neighbour (KNN) model.
   (c) Implementation and evaluation of a LightGBM (LGBM) model.
   (d) Implementation and evaluation of a Logistic Regression (Logit) model.
   (e) Implementation and evaluation of Naive Bayes (NB) models.
   (f) Implementation and evaluation of a Random Forest (RF) model.
   (g) Implementation and evaluation of a Support Vector Machine (SVM) model.
4. Comparison of the developed models.

## 1.3 Research Value

As outlined in Section 1.1, image classification is a field quickly growing in scope, with problems in everyday scenarios for both large companies and end users. There are many different approaches to solving image classification problems. As such, identifying the best method provides value for both further studies and for commercial users in reducing the time spent on model pre-screening and selection.

## 1.4 Roadmap

Each of the research objectives are explored in sections of this study. Section 2 examines the landscape of the field, as outlined in *Objective 1*. *Objective 2* is addressed in Section 3.2 where the preprocessing of the data is documented. Each of the modelling approaches outlined in objectives *3(a)* through *3(g)* are subsections of Section 5. Results are discussed in Section 5, addressing *Objective 4*.

# 2 Literature Review

In order to understand the outlook of the image classification field, a study of previous works ranging from seminal papers in the 1970s, to more contemporary papers of the 2020s. This study focused largely on papers published between 2012 and 2022, but acknowledgements to important earlier papers were included to better understand the evolution of the field.

## 2.1 Comparison of Previously Developed Models in Image Classification

Many studies have been conducted to determine and report on the effectiveness of a single model approach. Some notable studies are briefly outlined in Table 1 overleaf.

Table 1: Existing Models.

| **Model** | **Paper** | **Authors** | **Results** |
|---|---|---|---|
| SVM | A Relative Evaluation of Multiclass Image Classification by Support Vector Machines | Foody and Mathur 2004 | 87-93% Acc |
| SVM | Multi-Class Active Learning for Image Classification | Joshi, Porikli, and Papanikolopoulos 2009 | 70-94% Acc |
| SVM | Scalable active learning for multiclass image classification | Joshi, Porikli, and Papanikolopoulos 2012 | Up to 90% Acc |
| CNN | Systematic outperformance of 112 dermatologists in multiclass skin cancer image classification by convolutional neural networks | Maron et al. 2019 | 91% Specific 75% Sensitive |
| GAN (CNN) | Classification of hyperspectral images based on multiclass spatial-spectral generative adversarial networks | Feng et al. 2019 | 98.7-99.4% Acc |
| Logistic Regression, GBM, SVM | Deep learning framework for multi-class breast cancer histology image classification | Vang, Zhen Chen, and X. Xie 2018 | 77-83% Acc |
| ResNet (CNN) | Multi-class brain tumor classification using residual network and global average pooling | R. L. Kumar et al. 2021 | 97% average Acc. |
| Logistic Regression | Single-Label Multi-Class Image Classification by Deep Logistic Regression | Dong, Zhu, and Gong 2019 | 62-72% Acc. |
| Naive Bayes / Fusion CNN | Automatic crack recognition for concrete bridges by fully convolutional neural network and Naive Bayes data fusion based on visual detection system | Li et al. 2020 | 92-94% Acc. |

The papers in Table 1 show a preference for CNN based models in recent years. Results appear consistently higher with these models applied; however, this often comes with a longer training time and black box model, making it difficult or impossible to understand the process the model takes when coming to a classification decision. Some key papers from a more nascent period of the field prefer simpler models, often SVM, as is discussed in Section 2.2.

## 2.2 Image Classification

Image classification problems have been a prominent area of study in computer vision since the 1970s when researchers aimed at classifying images as examples of satellite photography, aerial photography, or microphotography (Haralick, Shanmugam, and Dinstein 1973). Early studies such as this typically focused on extracting textural features from images, notably smoothness and roughness. This has its issues. One such issue is in texture definition. What does it mean for an image to be "smooth" or "coarse"? Is it enough to suggest that a large variation in pixel values identifies coarseness, or are large contrasting blocks of pixels a better means of identifying rough patches; perhaps both are adequate (Lark 1996). Due to some of the issues presented in textural analysis, later studies examined different features in their analysis. One study focusing on colour showed the potential power of the feature when creating image segmentations (Belongie et al. 1998). Earlier, (Niblack et al. 1993) were among the first to use colour in classifying photographs. Niblack et al. also examined the shape of objects in the image using edge detection methods. Developments such as those introduced by these studies laid the foundations for future studies which examine a plethora more features, such as contextual features; spectromatic features; vegetation indices; post-transformation images; multi-temporal images; and ancillary data and metadata.

## 2.3 A Critical Review of Feature Selection

When choosing a means of classifying images, different features offer different levels of precision, and different results, as well as requiring different levels of pre-processing and a different shape of input. Commonly used features include texture, colour, shapes and metadata. However, these features can also be further delved into, introducing another degree of caution researchers must exhibit. As aforementioned, texture requires a clear and obvious definition in the data.

### 2.3.1 Shape

When classifying images often the analysis of the shapes of image parts provide a lot of information. Shape analysis is commonly used in classification of photographic images (Bosch, Zisserman, and Munoz 2007), spectral images (Blaschke 2003), and hyperspectral images (Mirzapour and Ghassemian 2015, Mercier and Lennon 2003).

When using shape as a feature it is also important to determine how shapes are defined. One method of shape detection is through edge detection, such as in (David 2020), or using "blobs" of colour as in (Belongie et al. 1998). Using edge detection techniques is common with Convolutional Neural Networks (S. Xie and Tu 2015), whereas simpler modelling techniques often utilise areas of similar colour. This analysis of colour blocks for identifying shape is the basis of the landmark face detection algorithm developed by Jones and Viola in the early 21st century (Jones and Viola 2003). Face detection algorithms have changed very little in the near 20 years since the article was published. This is due to the speed and accuracy of face detection by using areas of light and areas of shadow to identify faces. This simple algorithm can be looked to as an example indicating the power of colour blocks as a feature in object recognition.

### 2.3.2 Colour

Colour may take the form of one of many different colour spaces - or how the colour data is represented numerically. Common colour spaces are RGB, where Red, Green, and Blue pixel values are stored as a number between 0-255. HEX representation of a colour uses a base-16 number to identify colours. HSB represents the Hue, Saturation, and Brightness of pixels; YCbCr represents Luminance, Green-Blue difference, and Green-Red difference for pixels; CMYK uses the Cyan, Magenta, Yellow, and Black values for each pixel, among many others. The multitude of options means that researchers must make a careful choice when designing their model.

It was determined in a 2017 paper that choice of colour space can have a significant impact on the output of a model when detecting humans by skin tone (Kolkur et al. 2017). A 2018 study found differences in output of leukemia detection models when using CIELAB and CMYK colour spaces (Anilkumar, Manoj, and Sagi 2018).

Colour as a feature is very rarely used in isolation. When combined with other metrics, colour can be a powerful variable to include. Studies can often omit colour, with models frequently converting input images to greyscale as one of the early steps. The inclusion of colour can greatly increase the required processing power, as colour images can contain three to four times more data than black and white images. Colour sees greater use in object detection than image classification, as classification frequently uses greyscale images. One example of a colour based classification model appears in a paper by Ding et al. 2018.

### 2.3.3 Sub-pixel and Spectrographic Approaches

Image classification approaches are often taken in the field of remote sensing - the process of detecting the physical aspects of an area by measuring its reflected and emitted radiation at a long distance, such as from satellite imagery or aerial photography. These types of problems often introduce

mixed-pixels, pixel values which represent the energy output of an area. In reading the energy data for aerial photographs, one pixel will often represent an area with some variation in output. The accuracy of the data is frequently bottle-necked by the resolution of the images. Because of this, sub-pixel fluctuations are accounted for using stochastic models (Bosdogianni, Petrou, and Kittler 1994). These sub-pixel variations have proven to be a major issue in remote sensing classification models (Fisher 1997, Cracknell 1998).

Another frequently utilised approach in remote sensing is analysis of vegetation indices. The vegetation index of an area is determined by applying transformations to aerial images to produce images which focus on specific wavelengths of light. These transformations are then combined and adjusted to enhance the presence of green vegetation in the images. While this approach may not be useful in the analysis of images of the Martian surface, it has been successfully used in combination with other metrics in classification problems (Dai and Khorram 1998), indicating the viability of multi-modal models in image classification. Vegetation indices are used to enhance the green aspects of satellite imagery. However, a variation on the model to focus on other colour bands may be viable in different scenarios.

## 2.4 Identified Gaps and Conclusion

It is clear that feature selection plays an important role in classifying images based on their contents. Many studies have been conducted to examine the influence of colour space, shaping method, or aggregation approaches. However, a knowledge gap can be found in applying multiple models to the same images, and reviewing the same metrics on each. It is not clear whether any given model will produce the best results in these conditions. Studies often focus on one modelling approach, perhaps with a comparison to one previously favoured method. A full comparison of a breadth of modelling approaches appears to not exist in the image classification space.

# 3 Research Methodology and Design Specification

## 3.1 Data collection

Data were collected from the NASA Open Data Repository, an open source repository of data collected, used, or analysed by NASA JPL. The data are subject to a Creative Commons Attribution 4.0 International licence and are associated with the DOI 10.5281/zenodo.2538136.

## 3.2 Data Description and Preprocessing

### 3.2.1 Data Description

The data consist of 73,031 greyscale JPEG images with a resolution of 227x227 pixels. The images are labelled as featuring one of eight geological phenomena ('bright_dune', 'crater', 'dark_dune', 'impact_ejecta', 'other', 'slope_streak', 'spider', 'swiss_cheese'). The dataset consists of 10,433 unique images, each of which has undergone six transformations to increase the number of datapoints in the set and to increase the variety of images ensuring models have the ability to recognise these phenomena when photographed from different angles. The six transformations are:

- 90 degrees clockwise rotation
- 180 degrees clockwise rotation
- 270 degrees clockwise rotation
- Horizontal flip
- Vertical flip

- Random brightness adjustment

The distribution of the classes is imbalanced, with one of the eight classes featuring 83% of the data. Class representation can be seen in Table 1.

Table 2: Class distributions.

| index | proportion | count |
|---|---|---|
| bright_dune | 2.40% | 1,750 |
| crater | 6.71% | 4,900 |
| dark_dune | 1.56% | 1,141 |
| impact_ejecta | 0.32% | 231 |
| other | 83.60% | 61,054 |
| slope_streak | 3.19% | 2,331 |
| spider | 0.65% | 476 |
| swiss_cheese | 1.57% | 1,148 |

This imbalance is further addressed in section 3.2.3 below.

### 3.2.2 Justification

These data were chosen for the study as they are presented with an open licence; the images are aerial photographs which is a frequent scenario where image classification problems appear; they present a degree of novelty, with HiRISE data seeing little use in similar studies; and the data were recently published, uploaded to the NASA repository in December 2021. The degrees of availability, applicability, novelty, and recency were determined to be strong enough for use in a study such as this.

### 3.2.3 Preprocessing

Little preprocessing was done to the images themselves, but in order to be usable as an input to various models, amendments to the structure of the dataset were required. The data were sourced as a zip folder, containing

text and csv files, and a sub-folder which contained all 73,031 of the image files. The zip folder was decompressed. The text file included was a data description document. There were two csv files included; the first identified which of the classes each of JPEG files represented using a numeric encoding; the second contained a dictionary identifying which encoding was paired with which feature.

The images were grouped into folders, each folder representing one of the geological phenomena. This was to ensure the input for neural network models was organised matching the model's expectations. While this structure is standard for a neural model, other models (such as SVM) are more adaptable in input shape.

Before any of the data could be fed into a model, a change was made to the colour space. The images in the dataset are black and white from the source, however the encoding uses the RGB colour space. Each image was converted to a greyscale colour space. This was done in order to reduce the dimensionality of the image three-fold, thereby reducing its memory usage in kind. As the images were visually greyscale before this step, this conversion was lossless.

Models were generated using three different python modules - scikit-learn[1] (Pedregosa et al. 2011), LightGBM[2] (Ke et al. 2017) and tensorflow.keras[3] (Abadi et al. 2016). The TensorFlow models were trained using a TensorFlow Dataset object[4] which was generated using the TensorFlow function *image_dataset_from_directory*. The LightGBM and scikit-learn models were trained on a dataset which was generated using Principal Component Analysis (PCA). The dataset consisted of 15 components, covering 87.45% of the variance of the original pixel value dataset. The dataset contained 51% of the original data in order to minimise the heavy class imbalance. This subset consisted of all of the original data in seven classes (11,977 images), and 25,000 of the images from the *other* class. The values in the

[1] https://scikit-learn.org/stable/about.html

[2] https://lightgbm.readthedocs.io/en/v3.3.2/

[3] https://www.tensorflow.org/about

[4] https://www.tensorflow.org/api_docs/python/tf/data/Dataset

Table 3: Models and Frameworks.

| Framework | Model | Dataset |
|---|---|---|
| scikit-learn | K-nearest Neighbours | PCA |
| | Gaussian Naive Bayes | PCA |
| | Multinomial Naive Bayes | PCA |
| | Complement Naive Bayes | PCA |
| | Logistic Regression | PCA |
| | Random Forest | PCA |
| | Support Vector Machine | PCA |
| LightGBM | LightGBM | PCA |
| TensorFlow | Convolutional Neural Network | TensorFlow Dataset |

PCA DataFrame were scaled between 0 and 1 for better interaction with the models.

# 4 Research Methodology

Nine different modelling approaches were carried out. These approaches are outlined in Table 3 , identifying the framework and the dataset the model utilised.

# 5 Implementation and Evaluation

Each of the methods carried out were analysed using multiple metrics. Namely the accuracy, weighted average recall, F1 score, and Cohen's kappa value. Accuracy is the overall proportion of correct predictions. Recall is the fraction of the positive class which were correctly identified as positive. The F1 score is the harmonic mean of the precision and recall, where precision is the proportion of positive classifications that were correct. Cohen's kappa score is an accuracy score which has been adjusted for agreement

occurring by chance. Formulae for each metric can be seen below.

$$Accuracy = \frac{(TP+TN)}{(TP+TN+FP+FN)}$$

$$Recall = \frac{TP}{TP+FN}$$

$$F_1 = \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

$$\text{Kappa}(\kappa) = \frac{2 \times (TP \times TN - FN \times FP)}{(TP+FP) \times (FP+TN) + (TP+FN) \times (FN+TN)}$$

where $TP = TruePositive$, $TN = TrueNegative$, $FP = FalsePositive$ and $FN = FalseNegative$.

## 5.1 K-nearest Neighbours

The K-nearest neighbour (KNN) algorithm is used to classify datapoints into a category by taking the most frequent category of neighbours to that datapoint when plotted in *n* dimensional space, where *k* refers to the number of neighbours examined in classification.

**Implementation:** The K-nearest neighbours approach was carried out using a k-value of 5. This was due to the largely imbalanced dataset. Larger values for k inflated the likelihood of elements having a significant number of neighbours in the *other* class. K-nearest neighbours models were generated using the $KNeighboursClassifier$ class from the scikit-learn framework[5]

**Evaluation:** The low kappa values indicate a weak model, with minimal agreement (McHugh 2012). This indicates that while the accuracy of the model is in the 60-70% range, this is largely due to chance more so than the inherent strength of the model.

[5]https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

Table 4: K-nearest neighbour metrics.

| Metric | Score |
|---|---|
| Kappa | 0.28 |
| Accuracy | 66.66% |
| F1 | 0.6396 |
| Weighted Recall | 0.6666 |

## 5.2 Naive Bayes

The Naive Bayes method of classification is a probabilistic model, basing classifications on Bayes' theorem where probability of a given event is based on knowledge of conditions leading to the event. Bayes' theorem can be expressed as $P(A|B) = \frac{P(B|A)P(A)}{P(B)}$.

Three Bayesian approaches were carried out. A Gaussian approach; a multinomial approach; and a complement approach. The models were generated using the *GaussianNB*, *ComplementNB*, and *MultinomialNB* classes from the scikit-learn framework[6].

### 5.2.1 Gaussian Naive Bayes

A Gaussian Naive Bayes model allows for classification of a continuous variable. In this instance, the target variable can be encoded as a value between 0 and 1 to enable the use of a Gaussian approach. The values for a Gaussian classifier can be determined through the equation for a normal distribution:

$$P(x = v \mid C_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} exp\left(-\frac{(x_i - \mu_k)^2}{2\sigma_k^2}\right)$$

where $x$ is a continuous attribute, $\mu_k$ represents the mean of the values in $x$ associated with class $C_k$, and $\sigma_k^2$ is the Bessel corrected variance of the

[6]https://scikit-learn.org/stable/modules/naive_bayes.html

values in $x$ associated with class $C_k$.

**Implementation:** The model was created using the *GaussianNB* function from the scikit-learn framework. Default values were used when running the model and no priors were specified.

Table 5: Gaussian Naive Bayes metrics.

| Metric | Score |
|---|---|
| Kappa | 0.15 |
| Accuracy | 52.60% |
| F1 | 0.5372 |
| Weighted Recall | 0.5260 |

**Evaluation:** Once again, the kappa score is very low. Through McHugh's analysis, 0-4% of data produced by the model are reliable. This indicates that the predictions from the model are very weak. The accuracy score of 53% is not indicative of a strong model.

### 5.2.2 Multinomial Naive Bayes

A Multinomial Naive Bayes model is used where data is multinomially distributed. This is most useful in making predictions with multiple predictive variables.

The distribution of the data is specified by vectors $\theta_y = (\theta_{y1}, \ldots, \theta_{yn})$ for each class $y$, where $n$ is the number of features, and $\theta_{yi}$ is the probability $P(x_i \mid y)$ of feature $i$ appearing in a sample belonging to class $y$.

The parameters $\theta_y$ are estimated using a smoothed maximum likelihood formula: $\hat{\theta}_{yi} = \frac{N_{yi}+\alpha}{N_y+\alpha n}$ where $N_{yi} = \sum_{x \in T} x_i$ is the number of times feature $i$ appears in a sample of class $y$ in the training set $T$, and $N_y = \sum_{i=1}^{n} N_{yi}$ is the total count of all features for class $y$.

**Implementation:** The model was created using the *MultinomialNB* function from the scikit-learn framework. The data were fed into the model using the default arguments. Smoothing was applied with an $\alpha$ value of 1.

Table 6: Multinomial NB metrics.

| Metric | Score |
|---|---|
| Kappa | 0.00 |
| Accuracy | 67.61 |
| F1 | 0.5455 |
| Weighted Recall | 0.6761 |

**Evaluation:** The kappa score of 0 indicates that the model is completely unreliable. This is due to the model predicting every input to correspond to the *other* class. This is not surprising as multinomial models, while applicable to multi-class data, typically favour binary prediction.

### 5.2.3 Complement Naive Bayes

Complement Naive Bayes is an adaptation of a Multinomial Naive Bayes that is better suited to unbalanced datasets. As one class in the PCA transformed data accounts for 68% of the training data, it was posited that a CNB would be a better approach.

**Implementation:** The model was created using the *ComplementNB* function from the scikit-learn framework. As with the multinomial model, the data were fed into the model using the default arguments with smoothing applied at an $\alpha$ value of 1.

Table 7: Complement Naive Bayes metrics.

| Metric | Score |
|---|---|
| Kappa | 0.06113 |
| Accuracy | 22.54 |
| F1 | 0.2481 |
| Weighted Recall | 0.2254 |

**Evaluation:** Once more, a low kappa signifies the model's inability to reliably predict images from this dataset. A marginal improvement over the multinomial classifier can be seen but this is largely insignificant due to the

low scores. As the complement model is based on the multinomial model, it is also an expected result to have poor performance.

## 5.3 Logistic Regression

Logistic Regression is a classification algorithm which generates a probability of a datapoint belonging to a class. Typically used for binary classification, but through the scikit-learn framework and multinomial approach is possible.

**Implementation:** The model was created using the *LogisticRegression* class from the linear_model module of scikit-learn[7]. L2 was chosen as the penalty to apply. Scikit-learn allows multiple options when using a logistic regressor for multi-class data: *OvR* (One versus rest) or *multinomial*. Both methods were applied and the best results chosen. The reported results relate to the multinomial implementation.

Table 8: Logistic Regression metrics.

| Metric | Score |
|---|---|
| Kappa | 0.03154 |
| Accuracy | 67.58% |
| F1 | 0.55631 |
| Weighted Recall | 0.67577 |

**Evaluation:** The logistic model also produces a very poor kappa value. Agreement is near zero, once again indicating that the model produces correct classifications largely through chance.

## 5.4 Random Forest

A Random Forest is a collection and aggregation of multiple decision tree models, each trained on different subsets of the total training dataset.

[7]https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

**Implementation:** The model was produced using the *RandomForestClassifier* function from the scikit-learn framework[8]. The *Gini* impurity criterion was used to measure split purity. In all, 100 estimators were used in the model.

Table 9: Random Forest metrics.

| Metric | Score |
|---|---|
| Kappa | 0.2806 |
| Accuracy | 72.07 |
| F1 | 0.6629 |
| Weighted Recall | 0.7207 |

**Evaluation:** The Random Forest model produces a kappa value slightly better than most previous mentioned models, however the results are still considered to be poor, with minimal agreement.

## 5.5 Support Vector Machine

A Support Vector Machine (SVM) is a classifier that finds a hyperplane of the data and generates a decision boundary to categorise the data as inside or outside. Once again, an SVM is generally preferred for use in binary classification problems, but scikit-learn allows multiclass operations through *OvO* (One vs One) classification or *OvR* classification, as it does with logistic regression. Many previous studies have utilised SVM in image classification to great success; however this often takes the form of binary classification such as in Goh, Chang, and Cheng 2001, Rejani and Selvi 2009, Vijayarajeswari et al. 2019, and Murugan, Nair, and K. Kumar 2019.

**Implementation:** The model was generated using the *SVC* function of the *svm* class from the scikit-learn framework[9]. Simple hyperparamter tuning was carried out using the *GridSearchCV* function[10]. This tuning revealed a

[8] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

[9] https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

[10] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

C value of 10 and a gamma value of 0.001 to be ideal; however the results show that an SVM was not the correct classifier for these data.

Table 10: Support Vector Machine metrics.

| Metric | Score |
|---|---|
| Kappa | 0.2451 |
| Accuracy | 71.91% |
| F1 | 0.6457 |
| Weighted Recall | 0.7191 |

**Evaluation:** The results for the SVM show poor to moderate agreement between the predictions and the true classes. While a marginal improvement over previous models is visible, the results are still too poor to consider viable in application.

## 5.6 LightGBM

LightGBM is a node first decision tree based gradient boosting algorithm. As most decision tree algorithms use a depth first build, LightGBM is structured quite differently. LightGBM is generally faster to compute, and more memory efficient than other similar models[11].

**Implementation:** An LGBMClassifier object was used to create the model. In this instance, the default values were used for each parameter. This includes a learning rate of 0.1, and an objective of *multiclass*.

Table 11: LightGBM metrics.

| Metric | Score |
|---|---|
| Kappa | 0.3286 |
| Accuracy | 70.39% |
| F1 | 0.6735 |
| Weighted Recall | 0.7039 |

[11] https://lightgbm.readthedocs.io/en/latest/Features.html

**Evaluation:** The results of the lightGBM model show more promise than many of the previous models; however a kappa score of 0.33 is still too low to deem the model to be successful.

## 5.7 Convolutional Neural Network

**Implementation:** The neural network was generated using a TensorFlow Keras Sequential model. The shape of the model was as seen in Table 12. The model was run for 10 epochs. Due to a difference in the architecture of the framework, it was not possible to obtain an F1 score or a weighted recall score for the Tensorflow model. A SparseCategoricalCrossentropy[12] loss function was used in its place. The metrics for the validation set for each epoch of the neural network can be seen in Table 13. Table 14 contains the metrics for the training set.

Table 12: Convolutional Neural Network summary.

| Layer | Output shape | Num of Params |
|---|---|---|
| Rescaling/Normalisation | (None, 227,227,1) | 0 |
| Conv2D | (None, 227, 227, 16) | 160 |
| MaxPooling2D | (None, 113, 113, 16) | 0 |
| Conv2D | (None, 113, 113, 32) | 4,640 |
| MaxPooling2D | (None, 56, 56, 32) | 0 |
| Conv2D | (None, 56, 56, 64) | 18,496 |
| MaxPooling2D | (None, 28, 28, 64) | 0 |
| Flatten | (None, 50176) | 0 |
| Dense | (None, 128) | 6,422,656 |
| Dense | (None, 8) | 1,032 |

**Evaluation:** It can be seen that the neural model produces a moderate to strong kappa value, ranging from 0.49 to 0.63 on the validation set, and reaching values as high as 0.99 on the training set. The high value in training implies a degree of overfitting on the data; however, the validation re-

[12]https://www.tensorflow.org/api_docs/python/tf/keras/losses/SparseCategoricalCrossentropy

Table 13: Convolutional Neural Network metrics (Validation Data).

| Epoch | Kappa (V) | Accuracy (V) | loss (V) |
|---|---|---|---|
| 1 | 0.4986 | 88.55% | 0.3712 |
| 2 | 0.6080 | 89.65% | 0.3432 |
| 3 | 0.6011 | 90.38% | 0.4324 |
| 4 | 0.6270 | 90.04% | 0.4239 |
| 5 | 0.6437 | 90.55% | 0.6125 |
| 6 | 0.6384 | 90.65% | 0.6503 |
| 7 | 0.6034 | 90.01% | 0.8180 |
| 8 | 0.6430 | 90.37% | 0.7235 |
| 9 | 0.6331 | 90.68% | 0.8868 |
| 10 | 0.6262 | 89.98% | 0.9021 |

sults are evidence of a slightly better balanced model. The change in the loss value may be an indicator of a sub-optimal model. While the validation loss rises with each iteration, the training loss decreases. This may be an indicator that the learning rate for the model is too high, and requires further tuning.

Table 14: Convolutional Neural Network metrics (Training data).

| Epoch | Kappa (T) | Accuracy (T) | loss (T) |
|---|---|---|---|
| 1 | 0.4848 | 88.18% | 0.3844 |
| 2 | 0.6358 | 90.89% | 0.2791 |
| 3 | 0.7648 | 93.68% | 0.1852 |
| 4 | 0.8565 | 95.98% | 0.1167 |
| 5 | 0.9118 | 97.47% | 0.0732 |
| 6 | 0.9407 | 98.28% | 0.0493 |
| 7 | 0.9535 | 98.65% | 0.0407 |
| 8 | 0.9664 | 99.02% | 0.0303 |
| 9 | 0.9698 | 99.12% | 0.0276 |
| 10 | 0.9727 | 99.20% | 0.0259 |

The metrics for both the training and validation sets are visualised in Figure 1 overleaf.

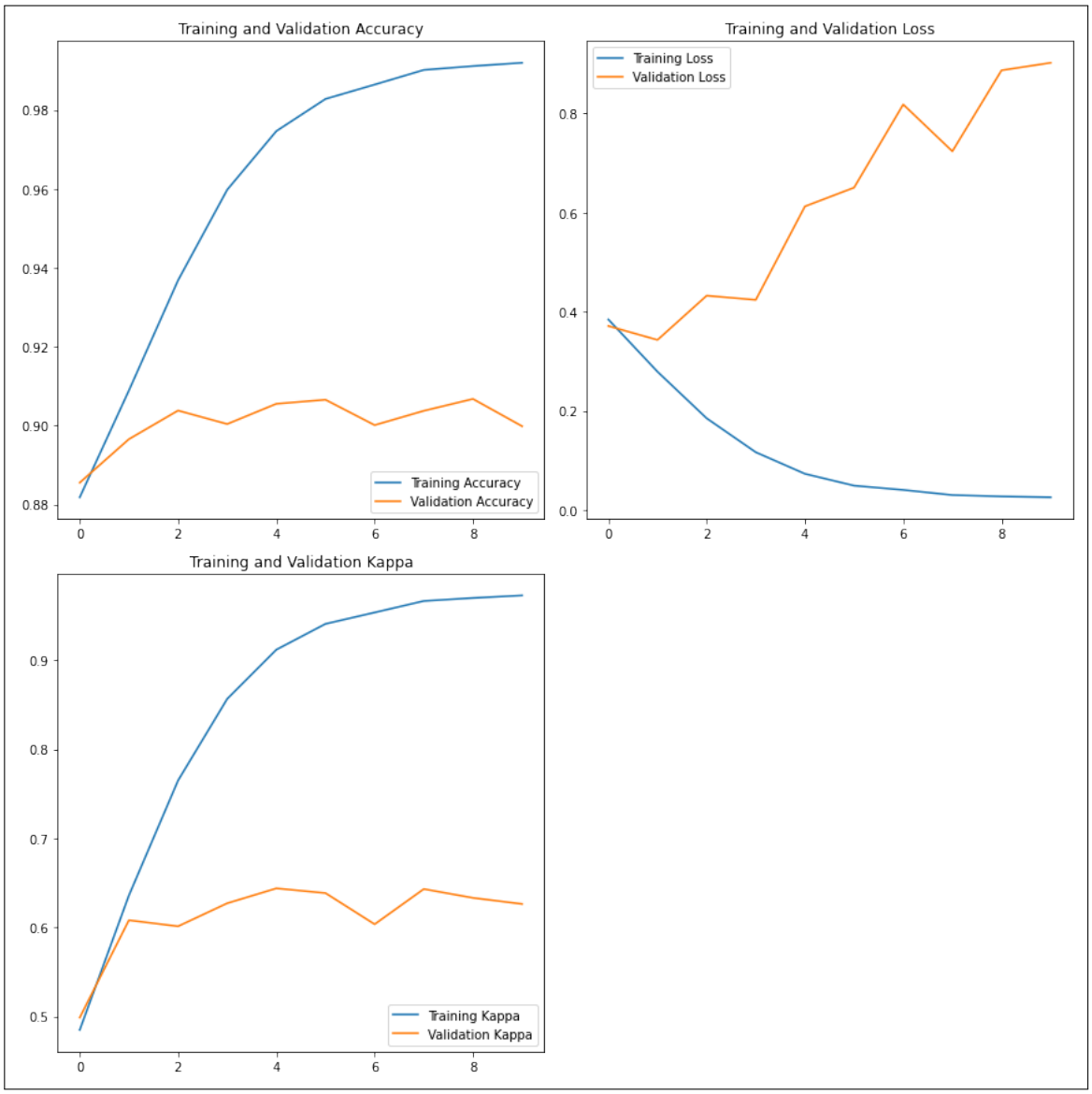



Figure 1: Metrics for training and validation sets of CNN.

## 5.8 Collated Results

For easier comparison, the results of each of the models can be seen in Table 15.[13]

Table 15: Collated metrics.

| **Technique** | **Kappa** | **Acc.** | **F1** | **Recall** |
|---|---|---|---|---|
| KNN | 0.28 | 66.66% | 0.64 | 0.67 |
| Gaussian NB | 0.15 | 52.60% | 0.54 | 0.53 |
| Multinomial NB | 0.00 | 67.62% | 0.55 | 0.68 |
| Complement NB | 0.06 | 22.54% | 0.25 | 0.23 |
| Logistic Regression | 0.03 | 67.58% | 0.56 | 0.68 |
| Random Forest | 0.28 | 72.07% | 0.66 | 0.72 |
| SVM | 0.25 | 71.92% | 0.65 | 0.72 |
| LightGBM | 0.33 | 70.39% | 0.67 | 0.70 |
| CNN (val) | 0.63 | 89.98% | N/A | 0.90 |

## 5.9 Discussion

It is clear when reviewing the metrics that the Convolutional Neural Network produced the most favourable results. In establishing this, we may accept an answer to our research question posed in Section 1.1. The question *"To what extent can direct comparisons between modelling techniques identify a clearly favourable method when applied to an image classification problem?""* appears to be answerable. With the above review, it appears that a convolutional neural network is the most effective model with some significance. This does come with some caveats however. It is worth noting that the device on which the study was conducted was power limited. It is possible that with more robust training and more in-depth and automated hyperparameter tuning, improvements to all models could occur. However due to the bottlenecks on the system, this would not be viable in a

[13] CNN results shown are from final epoch.

speedy manner. As such, little tuning was possible. This is a consideration that future works may address. This can also be seen in the method of data preprocessing used in many of the models. This technological constraint also created a limit where some models were unable to be generated using the image data itself, but rather required dimensionality reduction.

Principal Component Analysis was used as a dimensionality reduction technique to overcome technical barriers. More powerful machines may be able to train and fit models using the original data unaltered. As scikit-learn based models require a dataframe input, this would result in a dataframe with 73,032 rows and 51,530 columns, or approximately $3.6 \times 10^9$ data points. This proved too large a dataset, and as such dimensionality reduction was used to increase the digestibility of the data. Some information loss occurs in this dimensionality change.

# 6 Future Works and Conclusions

Future works may also consider the size and balancing of the dataset. The full dataset of 73,031 images while not insignificant, is likely too small to have produced the best models possible. The imbalance of one class containing 83% of images in the dataset also possibly hindered the development of some models. With the large number of instances in one of the classes, addressing this imbalance meant reducing the already small number of images used to train and fit the models. It can be determined that a larger dataset would address these issues.

While the results of this study show a CNN to be the preferred method of classification in images, these aforementioned concerns prevent certainty in this suggestion. A further consideration for developers designing an image classification system is time to train. The Convolutional Neural Network used in this study took approximately 3 hours to fully train, whereas each of the other models took minutes to train. This is largely due to the PCA dataset used by other models; however, it is expected that a CNN will take longer to train its weights and biases.

# 7 Acknowledgements

I would like to extend my sincerest thanks to everyone who supported me throughout the course of my studies. My wonderful partner Katie for her support and patience, as well as her proof-reading and error catching prowess; my friends for their help in keeping me grounded during a turbulent time and ensuring my breaks were not few and far between; and the staff at NCI, namely my supervisor Catherine Mulwa for their help and guidance.

# References

Abadi, Martin et al. (2016). "Tensorflow: A system for large-scale machine learning". In: *12th Symposium on Operating Systems Design and Implementation*, pp. 265–283.

Anilkumar, KK, VJ Manoj, and TM Sagi (2018). "Colour based Image Segmentation for Automated Detection of Leukaemia: A comparison between CIELAB and CMYK colour spaces". In: *2018 International Conference on Circuits and Systems in Digital Enterprise Technology (ICCSDET)*. IEEE, pp. 1–6.

Belongie, Serge et al. (1998). "Color-and texture-based image segmentation using EM and its application to content-based image retrieval". In: *Sixth International Conference on Computer Vision (IEEE Cat. No. 98CH36271)*. IEEE, pp. 675–682.

Blaschke, Thomas (2003). "Object-based contextual image classification built on image segmentation". In: *IEEE Workshop on Advances in Techniques for Analysis of Remotely Sensed Data, 2003*. IEEE, pp. 113–119.

Bosch, Anna, Andrew Zisserman, and Xavier Munoz (2007). "Image classification using random forests and ferns". In: *2007 IEEE 11th international conference on computer vision*. Ieee, pp. 1–8.

Bosdogianni, Panagiota, Maria Petrou, and Josef Kittler (1994). "Mixed pixel classification in remote sensing". In: *Image and Signal Processing for Remote Sensing*. Vol. 2315. SPIE, pp. 494–505.

Cracknell, Arthur P (1998). "Synergy in remote sensing-what's in a pixel?" In: *International Journal of Remote Sensing* 19.11, pp. 2025–2047.

Dai, Xiaolong and Siamak Khorram (1998). "A hierarchical methodology framework for multisource data fusion in vegetation classification". In: *International Journal of Remote Sensing* 19, pp. 3697–3701.

David, D (2020). "Retinal image classification system for diagnosis of diabetic retinopathy using morphological edge detection and feature extraction techniques". In: *Artech J. Eff. Res. Eng. Technol* 1, pp. 28–33.

Ding, Xintao et al. (2018). "Prior knowledge-based deep learning method for indoor object recognition and application". In: *Systems Science & Control Engineering* 6.1, pp. 249–257.

Dong, Qi, Xiatian Zhu, and Shaogang Gong (2019). "Single-label multi-class image classification by deep logistic regression". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 33. 01, pp. 3486–3493.

Feng, Jie et al. (2019). "Classification of hyperspectral images based on multiclass spatial–spectral generative adversarial networks". In: *IEEE Transactions on Geoscience and Remote Sensing* 57.8, pp. 5329–5343.

Fisher, Peter (1997). "The pixel: a snare and a delusion". In: *International Journal of Remote Sensing* 18.3, pp. 679–685.

Foody, Giles M and Ajay Mathur (2004). "A relative evaluation of multiclass image classification by support vector machines". In: *IEEE Transactions on geoscience and remote sensing* 42.6, pp. 1335–1343.

Goh, King-Shy, Edward Chang, and Kwang-Ting Cheng (2001). "SVM binary classifier ensembles for image classification". In: *Proceedings of the tenth international conference on Information and knowledge management*, pp. 395–402.

Haralick, Robert M, Karthikeyan Shanmugam, and Its' Hak Dinstein (1973). "Textural features for image classification". In: *IEEE Transactions on systems, man, and cybernetics* 6, pp. 610–621.

Jones, Michael and Paul Viola (2003). "Fast multi-view face detection". In: *Mitsubishi Electric Research Lab TR-20003-96* 3.14, p. 2.

Joshi, Ajay J, Fatih Porikli, and Nikolaos Papanikolopoulos (2009). "Multi-class active learning for image classification". In: *2009 ieee conference on computer vision and pattern recognition*. IEEE, pp. 2372–2379.

– (2012). "Scalable active learning for multiclass image classification". In: *IEEE transactions on pattern analysis and machine intelligence* 34.11, pp. 2259–2273.

Ke, Guolin et al. (2017). "Lightgbm: A highly efficient gradient boosting decision tree". In: *Advances in neural information processing systems* 30, pp. 3146–3154.

Kolkur, Seema et al. (2017). "Human skin detection using RGB, HSV and YCbCr color models". In: *arXiv preprint arXiv:1708.02694*.

Kumar, R Lokesh et al. (2021). "Multi-class brain tumor classification using residual network and global average pooling". In: *Multimedia Tools and Applications* 80.9, pp. 13429–13438.

Lark, RM (1996). "Geostatistical description of texture on an aerial photograph for discriminating classes of land cover". In: *International Journal of Remote Sensing* 17.11, pp. 2115–2133.

Li, Gang et al. (2020). "Automatic crack recognition for concrete bridges using a fully convolutional neural network and naive Bayes data fusion based on a visual detection system". In: *Measurement Science and Technology* 31.7, p. 075403.

Maron, Roman C et al. (2019). "Systematic outperformance of 112 dermatologists in multiclass skin cancer image classification by convolutional neural networks". In: *European Journal of Cancer* 119, pp. 57–65.

McHugh, Mary L (2012). "Interrater reliability: the kappa statistic". In: *Biochemia medica* 22.3, pp. 276–282.

Mercier, Grégoire and Marc Lennon (2003). "Support vector machines for hyperspectral image classification with spectral-based kernels". In: *IGARSS 2003. 2003 IEEE International Geoscience and Remote Sensing Symposium. Proceedings (IEEE Cat. No. 03CH37477)*. Vol. 1. IEEE, pp. 288–290.

Mirzapour, Fardin and Hassan Ghassemian (2015). "Improving hyperspectral image classification by combining spectral, texture, and shape features". In: *International Journal of Remote Sensing* 36.4, pp. 1070–1096.

Murugan, A, S Anu H Nair, and K Kumar (2019). "Detection of skin cancer using SVM, random forest and kNN classifiers". In: *Journal of medical systems* 43.8, pp. 1–9.

Niblack, Carlton Wayne et al. (1993). "QBIC project: querying images by content, using color, texture, and shape". In: *Storage and retrieval for image and video databases*. Vol. 1908. International Society for Optics and Photonics, pp. 173–187.

Papers With Code (May 2022). *Object Detection*. [Online; accessed 13. May 2022]. URL: https://paperswithcode.com/task/object-detection.

Pedregosa, F. et al. (2011). "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12, pp. 2825–2830.

Rejani, Y and S Thamarai Selvi (2009). "Early detection of breast cancer using SVM classifier technique". In: *arXiv preprint arXiv:0912.2314*.

ScienceDirect (May 2022). *Image Classification - an overview*. [Online; accessed 13. May 2022]. DOI: 10.1016/B978-1-78548-236-6.50002-7.

Vang, Yeeleng S, Zhen Chen, and Xiaohui Xie (2018). "Deep learning framework for multi-class breast cancer histology image classification". In: *International conference image analysis and recognition*. Springer, pp. 914–922.

Vijayarajeswari, R et al. (2019). "Classification of mammogram for early detection of breast cancer using SVM classifier and Hough transform". In: *Measurement* 146, pp. 800–805.

Xie, Saining and Zhuowen Tu (2015). "Holistically-nested edge detection". In: *Proceedings of the IEEE international conference on computer vision*, pp. 1395–1403.