Student Number: 164574 & Kaggle Team Name: Bedford FC

1. Approach

The machine learning approach that has been adopted into the classification task of determining if an image is sunny or not sunny is the Support Vector Machine (SVM). SVM's are a set of related supervised learning methods used for classification and regression. SVM's map the input vectors into a high dimensional feature space [1] via a kernel operator. In this high dimensional space, a maximal separating hyperplane is constructed and two parallel hyperplanes are constructed on each side of the separating hyperplane. The separating hyperplane performs classification by maximizing the distance between the two parallel hyperplanes and drawing a decision boundary. The vectors on one side of the separating hyperplane are labelled as -1 and all vectors on the other side are labelled as 1 [2].

2. Methodology

2.1. Model Selection

For the final classification model, the model uses a Support Vector Machine in python using the scikit-learn toolbox. Varieties of models were used initially for the classification task and SVM performed the best out of the 'out the box' classifiers. A Support Vector Machine (SVM) was used for this classification task because of its ability to easily perform binary classification, its capability for plotting each sample in n-dimensional space and the flexibility with the use of kernels to be optimised for the classification task.

2.2. Kernel Selection

A support vector machine has multiple available kernels such as Linear, Polynomial, RBF and Sigmoid that will change the function used. To optimise the accuracy of the support vector machine, each kernel was used to classify the same data. This resulted in the radial based function kernel becoming the most effective kernel. This is because the RBF kernel nonlinearly maps samples into a higher dimensional space unlike the linear kernel [3]. In the final classifier, a radial based function is used due to it providing the most optimal performances with high dimensionality.

2.3. Preprocessing

2.4. NaN Values

The additional training data contains NaN values across the 4608 features. While these values are missing within the features, the predictions are provided. The NaN values are replaced using the mean across the entire data set to make the additional training data useful. After the NaN values have been replaced, the annotated training data and the additional training data are concatenated, therefore increasing the amount of samples in the training data.

2.5. Scaling

Feature scaling is performed to standardise the range of features as the dataset contains features highly varying in range. Min-max scaling standardises the data between zero and one and the formula is illustrated in [4, *Figure 1*].

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Figure 1: Feature Scaling

2.6. Under sampling

In the concatenated training data, there are 1550 predictions with the value '1' and 1035 predictions with the value '0'. Due to the imbalance of the training data, undersampling is performed to balance the classes. This is achieved using imblearn under_sampling [5] and it undersamples the '1' value to make the classes equal size with a total size of 2070. When samples are removed, their corresponding annotated confidence value is also removed, therefore maintaining the integrity of the mapped sample predictions to the annotated confidence. Achieving this allows the annotated confidence labels to be used at classification as sample weights.

2.7. Feature Selection and data transformation

On the initial approach to solving the classification problem, there were attempts to complete extensive feature selection. The CNN and GIST features were separated and methods such as Principal Component Analysis were applied to reduce dimensionality. Separating CNN features was beneficial to the classification and produced increases in cross validation scores however, feature selection methods on GIST features were detrimental to the overall performance of the classifier and its cross validation scores. The GIST features were deemed important for the classification task and other methods such as Independent Component Analysis (ICA) were explored. ICA for pre-processing transforms the feature space into a feature space where the components are independent. ICA does not compress the features however; it removes correlations and higher order dependence [6].

3. Results and Discussion

3.1. Model

The highest performing model was the SVM, which combines all of the methodology described above.

3.2. Cross Validation

To measure accuracy and evaluate the performance of the classifier, k-fold cross validation (where k = 10) is performed. This is achieved using sklearn cross_val_score [7]. This returns a list of cross validation scores for each fold and the mean of the list is calculated to give a k fold cross validation score.

3.3. *Learning Curve*

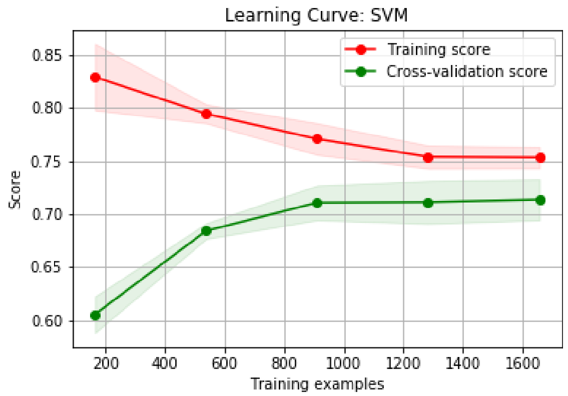

Figure 2: Learning Curve

Figure 2 illustrates the learning curve for the model. The learning curve demonstrates the classifier learning as the cross-validation score increases when training examples increase in size. *Figure 2* demonstrates a high bias as the training score and cross-validation score converge to a low score. A high bias means the classifier is more likely to make incorrect predictions and is not taking into account all the information in the data as it is under fitting. A solution to high bias is generating a more complex model as inserting more data will not improve the accuracy of the classifier [8].

3.4. Additional training data

	SVM	SVM with additional training data
Accuracy	0.58%	0.71%

Figure 3: Accuracy of model

In *Figure 3*, the SVM with the training data and the SVM with the training data and additional training data have the same pre-processing and classification applied. Both SVM's use the annotated confidence for sample weights in the classifier. The accuracy is calculated using cross validation scores with a 10 fold cross validation. As seen in *Figure 3*, the SVM with the additional training data performs significantly better than without. Using the additional training data in the classification model lead to the largest increase in accuracy and was the most important pre-processing task.

3.5. Annotated confidence

	SVM	SVM with Annotated confidence
Accuracy	0.709%	0.71%

Figure 4: Accuracy of model

In *Figure 4*, both SVM's include the additional training data and have the same pre-processing and classification applied. The accuracy is calculated using cross validation scores with a 10 fold cross validation. As seen in *Figure 4*, the difference between using the annotated confidence for sample weights and not using them is infinitesimally small. The annotated confidence is still used in the final classification model as it does show slight improvements in cross validation scores.

3.6. Discussion

Overall, the goals of the classification task were achieved. The model uses a SVM and this was optimised through grid searching techniques. The pre-processing goals were achieved as the data is scaled, under sampled and methods of feature extraction and data transformation have been applied.

If the classifier were to be improved to achieve better results, other methods of feature selection and dimension reduction would have been explored to yield stronger accuracies. RFECV [9] was briefly explored in the model however; it was not well optimised and did not provide ideal results. Other methods to improve the classifier would to be to investigate and provide solutions for the domain adaptation problem [10]. In the classification task, the domain adaptation problem is where the training data is seaside sceneries and the test data is urban sceneries.

As previously stated in the report, attempts were made to split the CNN and GIST features. While CNN features performed well, the GIST features did not. If the model were to be improved, it would be important to understand and investigate the importance of the features and how to pre-process the GIST features. These improvements would provide solutions to the high bias as the model would increase in complexity.

References

[1] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[2] K.-S. Goh, E. Chang, and K.-T. Cheng, "SVM binary classifier ensembles for image classification," *Proceedings*

of the tenth international conference on Information and knowledge management - CIKM01, 2001.

[3] D. Srivastava, “Data Classification using Support Vector.”

[4] G. Ciaburro, “Regression Analysis with R,” *O'Reilly | Safari*. [Online].Available: https://www.oreilly.com/library/view/regression-analysis-with/9781788627306/6bb0d820-6200-4bfe-aa91-e7b7ffa2a9c1.xhtml. [Accessed: 20-May-2019].

[5] “imblearn.under_sampling.RandomUnderSampler,” *imbalanced*. [Online]. Available: https://imbalanced-learn.readthedocs.io/en/stable/generated/imblearn.under_sampling.RandomUnderSampler.html#imblearn.under_sampling.RandomUnderSampler. [Accessed: 20-May-2019].

[6] V. Sanchez-Poblador, E. Monte-Moreno, and J. Solé-Casals, “ICA as a Preprocessing Technique for Classification,” *Independent Component Analysis and Blind Signal Separation Lecture Notes in Computer Science*, pp. 1165–1172, 2004.

[7] “sklearn.model_selection.cross_val_score,” *scikit*. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.cross_val_score.html. [Accessed: 20-May-2019].

[8] C. Perlich, “Learning Curves in Machine Learning,” *Encyclopedia of Machine Learning and Data Mining*, pp. 708–711, 2017.

[9] “sklearn.feature_selection.RFECV,” *scikit*. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFECV.html. [Accessed: 20-May-2019].

[10] W. M. Kouw, L. J. P. van der Maaten, J. H. Krijthe, and M. Loog, “Feature-Level Domain Adaptation,” *Journal of Machine Learning Research*, 01-Jan-1970. [Online]. Available: http://jmlr.org/papers/v17/15-206.html. [Accessed: 20-May-2019].