Land Cover Classification and Identification_Final Report

MUSA-650- Machine Learning in Remote Sensing

Xintian (Stella) Li, David (Yuan Rong) Pan, & Jacey Chang
May 9, 2021

1. Introduction:

Due to human activities, land cover is changing rapidly over the recent decades. Monitoring land cover changes is meaningful for better understanding and evaluating the urban development process and its associated environmental impacts. The proliferation of satellite image data and the emergence of advanced machine learning and deep learning technologies make it possible to identify land cover and quantify the changes automatically[1]. In this project, we will apply various machine learning and deep learning algorithms to EuroSAT, a European land cover dataset, to classify different categories of land cover. We will also compare the performance of our models on solving our particular land classification problem. This study tries to figure out which models perform better and offer insights on future land cover classification using satellite images.

2. Data:

We used the EuroSAT Dataset, a dataset created from Sentinel-2 satellite images, which are openly and freely accessible. The original EuroSAT images are tagged with 10 labels: Industrial Buildings, Residential Buildings, Annual Crop, Permanent Crop, River, Sea and Lake, Herbaceous Vegetation, Highway, Pasture, and Forest.

Among the 10 classes, some of the labels are branches of the same category, such as Industrial Buildings & Residential Buildings. As we are more interested in general land-use types, therefore, we combined similar labels. We grouped “AnnualCrop”, “PermanentCrop” and “Pasture” into “Agriculture”; “River” and “SeaLake” to “Water”; and “Residential” and “Industrial” to “Building”. Then we created a new balanced data set from the original EuroSAT.

Our new dataset has 6 classes, which are Agriculture, Forest, Herbaceous Vegetation, Highway, Building, and Water. Each class has 2,400 images with a size of 64 pixels by 64 pixels with 3 channels (the Red, Green, Blue channels). Examples of each label are shown in Figure 1.

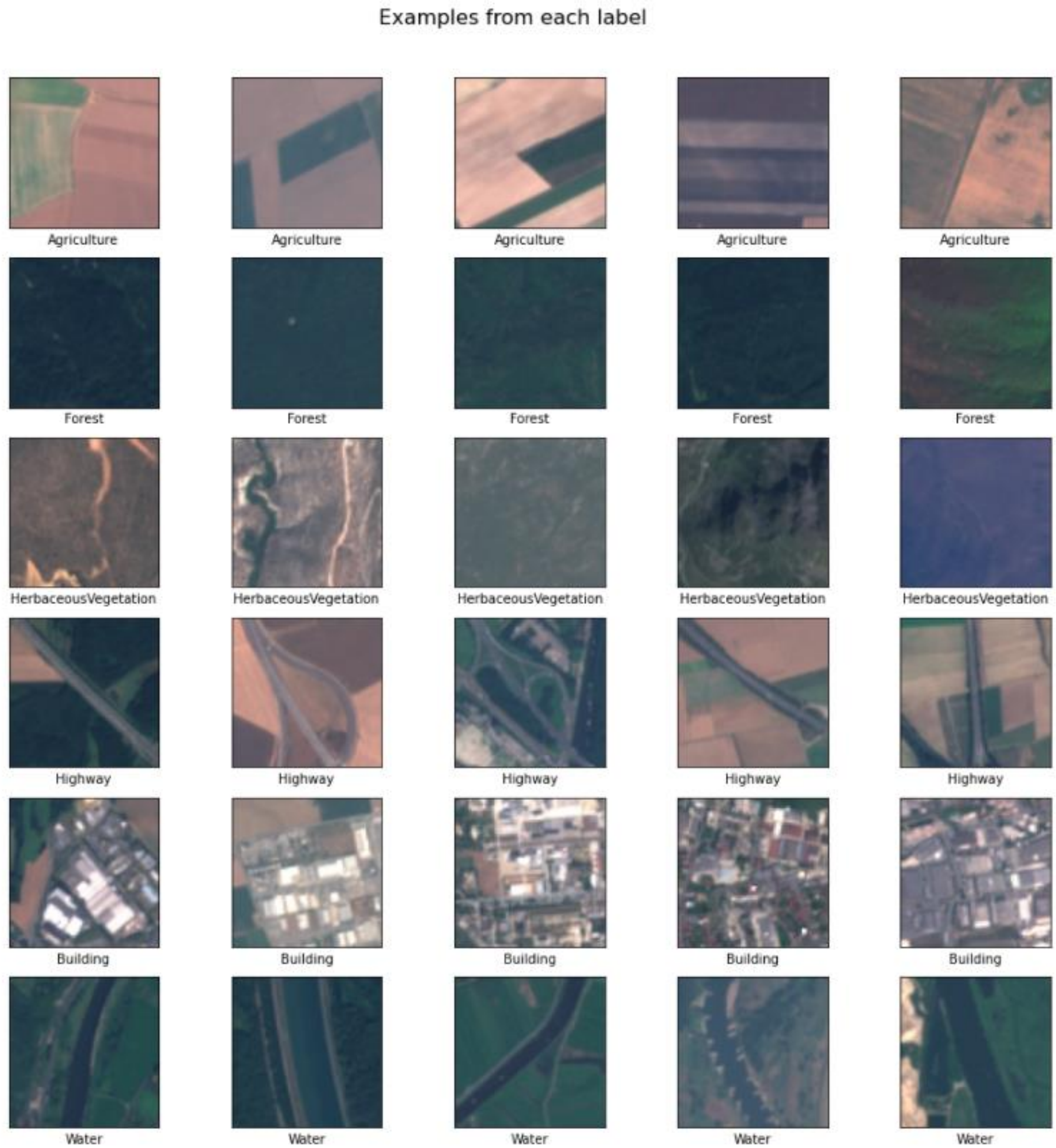


Figure 1: Examples from each label

3.Methods:

In this research, we applied both conventional machine learning methods such as K-means, SVM, and Random Forest; and deep learning methods, such as CNN, and transfer learning. We did Principal Component Analysis to reduce the dimensionality of the data before doing the machine learning models. We also tune the parameters for each model. Finally, we compared the performance between the different models. For the machine learning models, we used the modules in the scikit-learn package. For the deep learning models, we use Keras, the

TensorFlow's API, as well as other dependencies such as cudnn (7.6.5), cudatoolkit (>=9.0,<9.1), for GPU-acceleration.

3.1 Machine Learning Algorithms

Machine learning is a type of Artificial Intelligence that provides computers with the ability to learn without being explicitly programmed. It provides various techniques that can learn from and make predictions on data. There are three basic learning approaches in machine learning, which are supervised learning, unsupervised learning, and reinforcement learning[1]. In this research, we will mainly focus on supervised learning and unsupervised learning.

3.1.1 Data Preprocessing -- PCA

We first tried to run an SVM model using all 12288 dimensions for each sample, with a 70/30 training-test split. However, the model took more than several hours to run, making it difficult to do further cross-validation and parameter tuning. Therefore, we decided to reduce dimensionalities for each sample.

We ran PCA (Principal component analysis) as our main method for dimension reduction. Surprisingly, the first PC was of crucial importance, which explains 64% of the variance in the data. As is shown in Figure 2, the first 30 PCs could explain over 82% of the variance and the first 100 PCs could explain about 88% of the data variance.

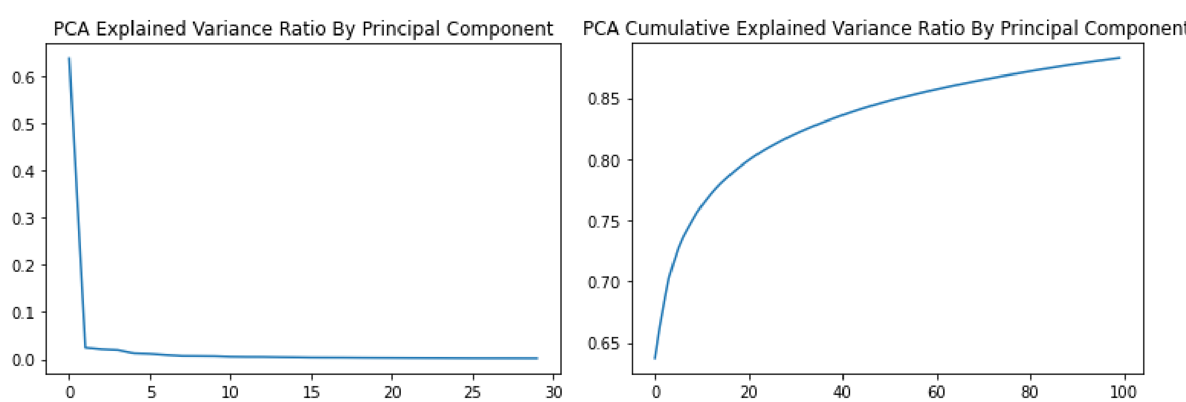


Figure 2: PCA explained variance ratio

[1] Lukas Masuch, "Deep Learning - A Visual Introduction," https://www.slideshare.net/LuMa921/deep-learning-a-visual-introduction.

We also plotted the principal components to see if there was any useful information (Figure 3). It seemed that the average hue of the images matters, especially the “yellowness” in the images (as shown in the first principal component). We thought this was reasonable because images of different classes do tend to have different colors.

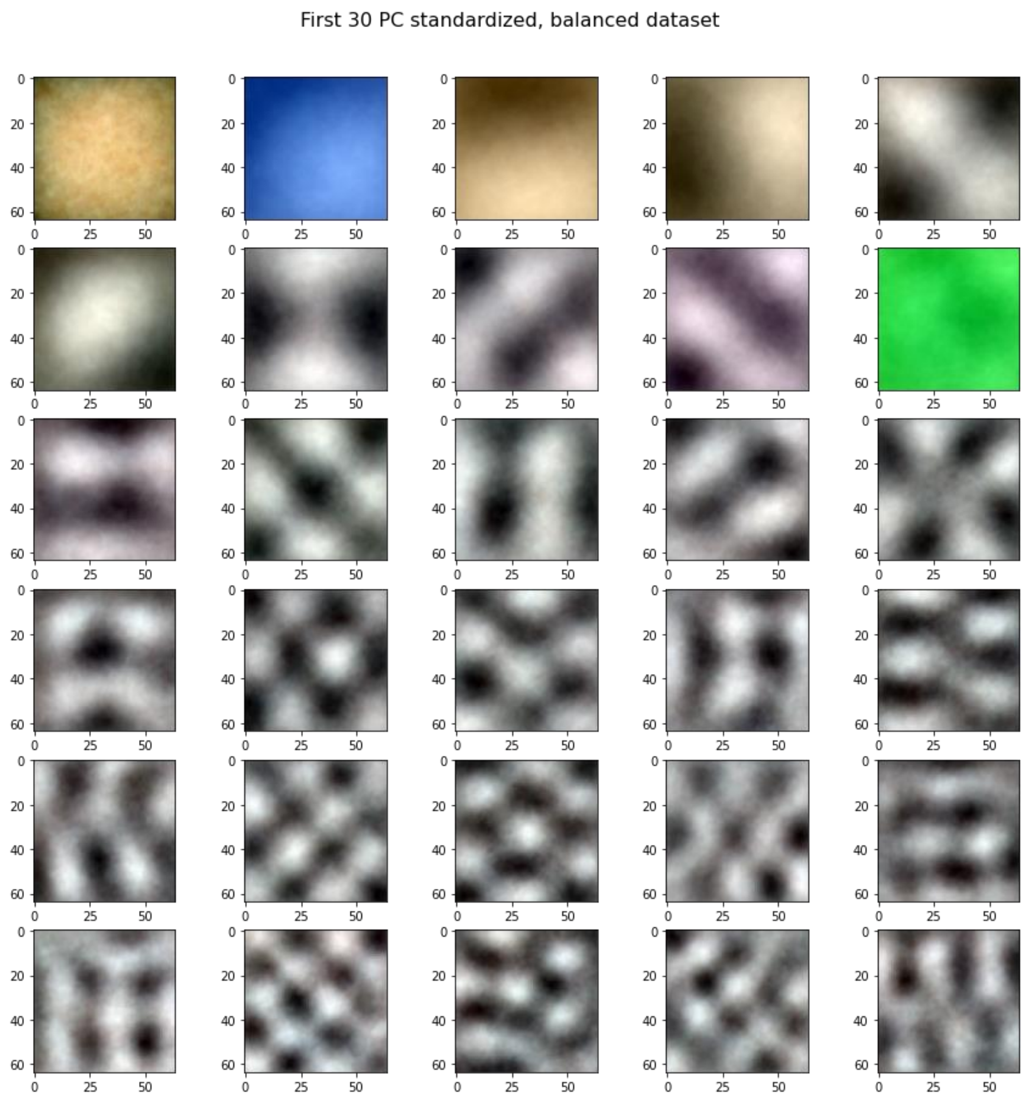


Figure 3: Principal component visualization

Finally, we decided to use 30 PCs in our following machine learning models.

3.1.2 KMeans

First, we used KMeans, an unsupervised learning algorithm, to see whether the images could be divided into 6 separate clusters without learning their labels. We choose this method because we want to study the performance of unsupervised learning in satellite image classification, and compare it with supervised learning algorithms. KMeans is a clustering method that discovers patterns in unlabeled data[2]. As a result, the model will only divide the samples into 6 clusters, without specifying any correlations between the clusters and the original labels. The criterion for good performances of KMeans will be that images of each label are divided into a separate distinct cluster, without being mixed up with other labels.

3.1.3 SVM and random forest classifiers

SVM is a supervised machine learning method commonly used to classify (linearly separable) binary problems by searching for a hyperplane (in different dimensions) in the data. For multi-class problems, it expands into multiple binary classification cases.

Random forest is a supervised ensemble learning method for either classification (or regression) that, based on the given images and classes, automatically selects the best predictive model out of a decision tree of models.

We split the balanced dataset with 70% as training data and 30% as testing data and used cross-validation to make full use of the training data. To tune the parameters, we applied grid search and used the mean test score in cross-validation as the criterion to choose parameters. We narrowed down the range of search each time to get the best parameters. Lastly, we calculated the accuracy of prediction on the test set.

3.2 Deep Learning Algorithms

3.2.1 CNN (Binary & Multiclass)

CNN is a deep learning algorithm that learns convolutional filters from input data automatically and identifies and filters all the important features from the data. We first tried using this algorithm to solve a simple 2-label classification problem (building and water, which are very distinct). Each of the two categories has a balanced subset of 2400 images. Then this was run for the multi-class land use classification (6 labels), again with each label having a balanced subset.

---

[2] Masuch.

3.2.2 Transfer Learning Models

To further our model comparisons, we imported two well-known pre-trained models (VGG16 and MobileNetV2) with relatively lower depths to fit and test on our own datasets. We set input shape to (64, 64, 3), froze the convolutional layers of the pre-trained model, transferred to our datasets, and updated the dense layers to get the output labels.

We will have used plots of validation and training accuracy for each epoch to evaluate the model performance. Confusion matrices indicating accuracy for predicting each individual label were also created. A ROC curve was made for the binary classifier to illustrates its diagnostic ability[3]. For the multi-class CNN, we have used categorical accuracy as the metric instead of general accuracy so that it is compatible with our one-hot encoding of the multiple categories,while retaining the same intuitiveness.

4. Results:

4.1 Machine Learning Model Results:

4.1.1 K-Means:

We found that KMeans model did a poor job distinguishing different classes correctly. Although 80% of the forest images were divided into cluster 5, the model failed in classifying other land covers. Therefore, unsupervised clustering is not an effective model in classifying satellite image patches.

cluster	0	1	2	3	4	5
actual_label						
Agriculture	129	219	146	53	166	7
Building	202	24	170	67	257	0
Forest	8	136	0	0	0	576
HerbaceousVegetation	242	161	124	18	151	24
Highway	235	232	38	41	135	39
Water	160	235	4	0	102	219

Figure 4: The cross-table for K-means Clustering

[3] "Simple Guide on How to Generate ROC Plot for Keras Classifier | DLology," accessed May 9, 2021, https://www.dlology.com/blog/simple-guide-on-how-to-generate-roc-plot-for-keras-classifier/.

4.1.2 SVM:

For the SVM model, we used OneVsRest SVM Classifier, since there are multiple labels in the dataset. The best SVM model we got was the one with rbf kernel and C=15. The testing accuracy of our best SVM model was 67%.

	params	mean_test_score	std_test_score	rank_test_score
2	{'estimator__C': 15, 'estimator__kernel': 'rbf'}	0.671429	0.012084	1
3	{'estimator__C': 20, 'estimator__kernel': 'rbf'}	0.669841	0.011362	2
4	{'estimator__C': 30, 'estimator__kernel': 'rbf'}	0.669544	0.011607	3
1	{'estimator__C': 10, 'estimator__kernel': 'rbf'}	0.669345	0.009516	4
0	{'estimator__C': 1, 'estimator__kernel': 'rbf'}	0.629266	0.012061	5

Figure 5: Rank of SVM models with different parameters when doing the cross validation

predicted_label	Agriculture	Building	Forest	HerbaceousVegetation	Highway	Water
actual_label						
Agriculture	388	67	26	125	83	31
Building	21	551	1	75	58	14
Forest	14	0	696	1	0	9
HerbaceousVegetation	76	85	16	472	49	22
Highway	95	110	12	69	338	96
Water	46	37	75	18	89	455

Figure 6: The confusion matrix for the best SVM

4.1.3 Random Forest:

The random forest model achieved a highest testing accuracy of 68.5% after parameter optimization (although the training accuracy was 100%). The best setting was 400 estimators and square root of features as the number of features to consider in the model. Similar to the results in K-Means the RF model was most accurate in predicting “Forest”, as opposed to other labels, as observed from the confusion matrix below.

	params	mean_test_score	std_test_score	rank_test_score
8	{'max_features': 'auto', 'n_estimators': 400}	0.680357	0.007448	1
1	{'max_features': 'sqrt', 'n_estimators': 200}	0.678869	0.007517	2
5	{'max_features': 'log2', 'n_estimators': 400}	0.677282	0.011021	3
2	{'max_features': 'sqrt', 'n_estimators': 400}	0.676984	0.010959	4
7	{'max_features': 'auto', 'n_estimators': 200}	0.675794	0.008566	5
4	{'max_features': 'log2', 'n_estimators': 200}	0.672024	0.009893	6
6	{'max_features': 'auto', 'n_estimators': 100}	0.670734	0.007979	7
3	{'max_features': 'log2', 'n_estimators': 100}	0.667956	0.013266	8
0	{'max_features': 'sqrt', 'n_estimators': 100}	0.666865	0.012826	9

Figure 7: The rank of random forest using different parameters

predicted_label	Agriculture	Building	Forest	HerbaceousVegetation	Highway	Water
actual_label						
Agriculture	445	81	19	93	63	19
Building	13	594	0	60	42	11
Forest	19	0	682	2	0	17
HerbaceousVegetation	108	93	7	427	65	20
Highway	122	137	4	42	384	31
Water	45	52	37	17	143	426

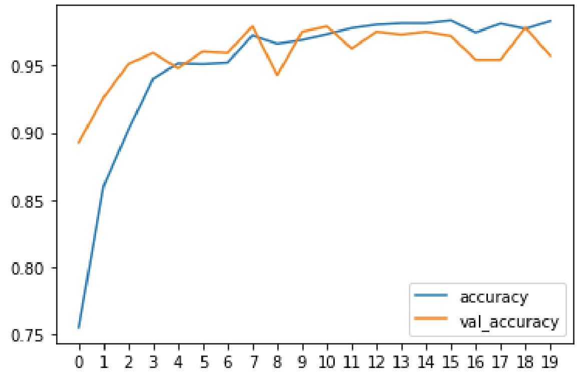
Figure 8: The confusion matrix for the best random forest model

The performances of the SVM and random forest classifiers were similar. We also noticed that the parameter tuning process didn't significantly improve the performance of either SVM or random forest model (the accuracies appear to be plateaued). Therefore, to get better accuracies, we have used deep learning models.

4.2 Deep Learning Model Results

4.2.1 CNN (binary):

The test accuracy of the two-label classification CNN model was 96% using only 1 convolutional layer and 1 dense layer. As you can see from the confusion matrix and the ROC (top-left curve), labels were overall correctly predicted (i.e. 1 as 1, 0 as 0).



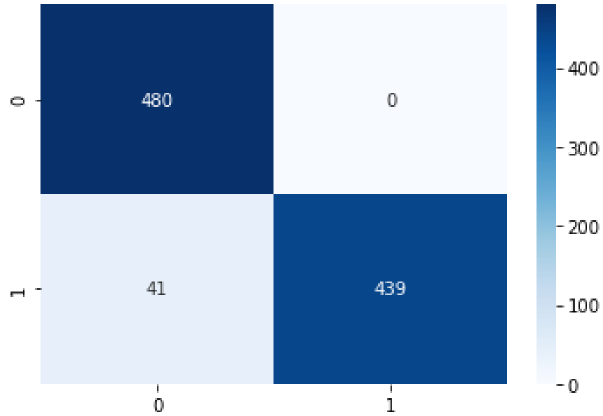


Figure 9: Training and validation accuracies for each epoch. Figure 10: The confusion matrix for the binary CNN classifier

The ROC curve curves towards the top left, indicating the good diagnostic ability of the binary classifier.

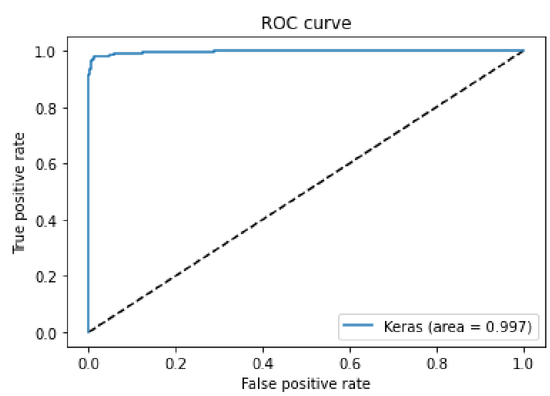


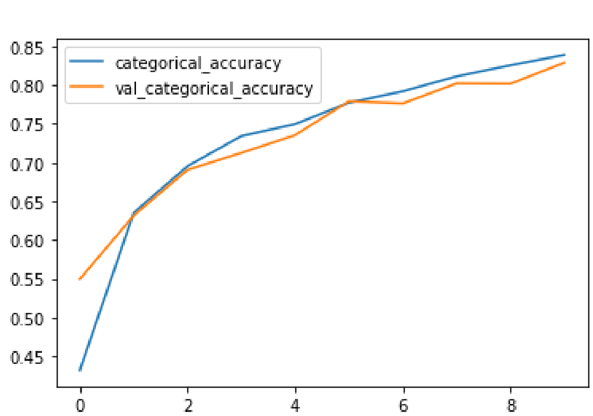
Figure 11: The ROC curve for the binary CNN classifier

```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_8 (Conv2D)            (None, 64, 64, 32)        896
_________________________________________________________________
max_pooling2d_8 (MaxPooling2 (None, 32, 32, 32)        0
_________________________________________________________________
dropout_10 (Dropout)         (None, 32, 32, 32)        0
_________________________________________________________________
flatten_2 (Flatten)          (None, 32768)             0
_________________________________________________________________
dense_4 (Dense)              (None, 100)               3276900
_________________________________________________________________
dropout_11 (Dropout)         (None, 100)               0
_________________________________________________________________
dense_5 (Dense)              (None, 2)                 202
=================================================================
Total params: 3,277,998
Trainable params: 3,277,998
Non-trainable params: 0
_________________________________________________________________
```

Figure 12: The layers used in 2-label CNN classifier

4.2.2 CNN (multi-class):

The multi-class classification (6 labels) has a test accuracy of 83%. As is shown in the confusion matrix (Figure 14), when predicting the 6 labels, the multi-label CNN classifier is best at predicting Label 1, which is the Forest (Same as the case when using other machine learning models). It is also relatively good at predicting Herbaceous Vegetation, Highway and Building. The model did not perform very well when predicting Agriculture patches and Water patches. One reason could be that we combined three subcategories into Agriculture, which are Annual Crop, Permanent Crop, and Pasture. Also, it is interesting to note that 106 of the water samples are mistakenly labeled as Forest. It might be because that forest and sea share some similar textures/distinguishing features.



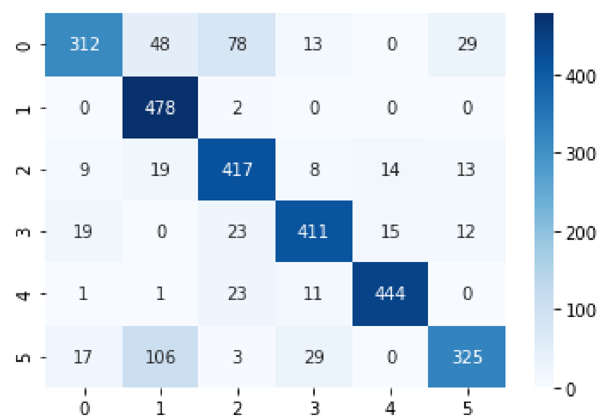


Figure 13: Training and validation accuracies of multi-class CNN. Figure 14 (right): The confusion matrix for the multi-class CNN classifier

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 64, 64, 32)        896
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 32, 32, 32)        0
_________________________________________________________________
dropout (Dropout)            (None, 32, 32, 32)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 32, 32, 64)        18496
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 16, 16, 64)        0
_________________________________________________________________
dropout_1 (Dropout)          (None, 16, 16, 64)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 16, 16, 128)       73856
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 8, 8, 128)         0
_________________________________________________________________
dropout_2 (Dropout)          (None, 8, 8, 128)         0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 8, 8, 256)         295168
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 4, 4, 256)         0
_________________________________________________________________
dropout_3 (Dropout)          (None, 4, 4, 256)         0
_________________________________________________________________
flatten (Flatten)            (None, 4096)              0
_________________________________________________________________
dense (Dense)                (None, 120)               491640
_________________________________________________________________
dropout_4 (Dropout)          (None, 120)               0
_________________________________________________________________
dense_1 (Dense)              (None, 6)                 726
=================================================================
Total params: 880,782
Trainable params: 880,782
Non-trainable params: 0
_________________________________________________________________
```

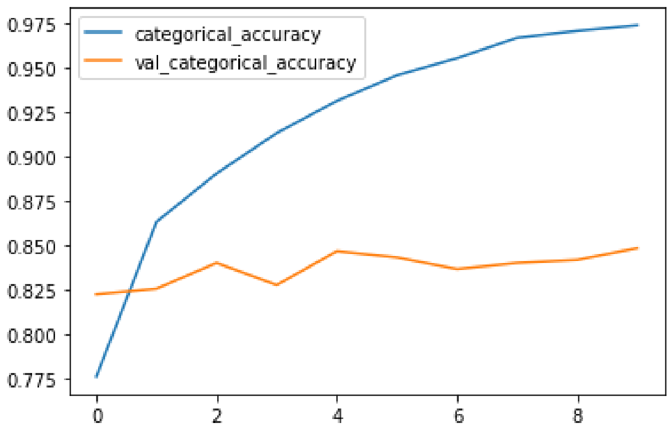
Figure 15: The layers and structure of the multi-label CNN classifier

By comparing the 2 CNN models compiled by ourselves, we noticed that CNN model performs better when predicting 2-label classification. It is reasonable because the 2-label question is much easier, given that we select two very distinct classes, building and water.

Also, we found that the accuracies of both CNN classifiers are much higher relative to the non-CNN classifiers such as Random Forest, SVM and K-Means; and was expected. We could argue that CNN models and deep learning algorithms have more predicting powers when classifying satellite images and doing land cover detections because they are able to, for instance, take into account the neighbor information in the images (no loss of information due to vectorization).

4.2.3 Transfer Learning Model 1 (MobileNetV2):

MobileNetV2 shows a testing accuracy (val_categorical_accuracy) of 84.8%, despite the convolutional layers being frozen, which shows that this pre-trained model is very well trained across all kinds of image categories, and may be adequately used for land use classification applications. The confusion matrix shows high predictive accuracies for forest and building classes, and weaker for those of agriculture and highway.



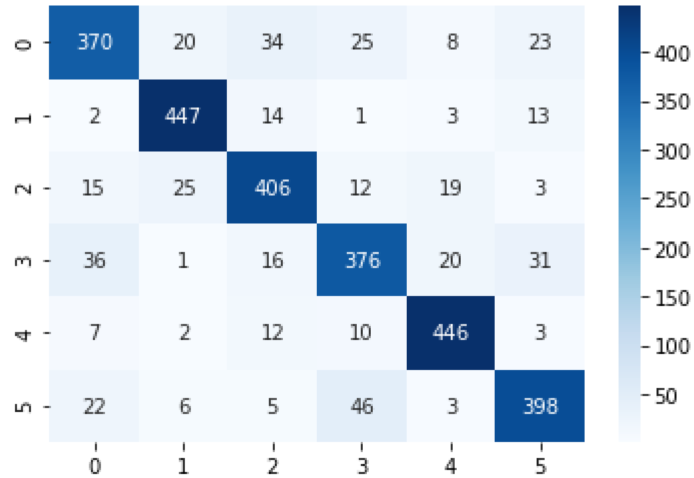
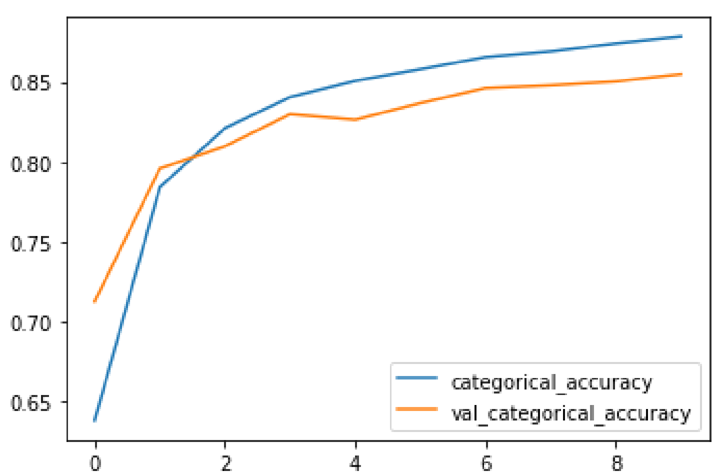


*Figure 16: Training and validation accuracies of **MobileNetV2**. Figure 17 (right): The confusion matrix for the **MobileNetV2***

4.2.4 Transfer Learning Model 2 (VGG16):

VGG16 mirrored the results of MobileNetV2, with a testing accuracy (val_categorical_accuracy) of 85.5%, despite the convolutional layers being frozen, which, again shows that this pre-trained model is very well trained across all kinds of image categories and may be adequately used for land use classification applications. The confusion matrix again shows high predictive accuracies for forest and building classes, and weaker for those of agriculture and highway.



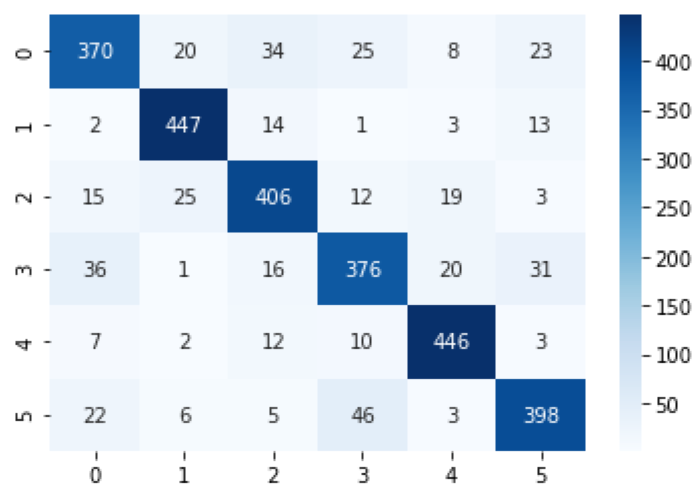


*Figure 18: Training and validation accuracies of **VGG16**. Figure 19 (R): The confusion matrix for the **VGG16***

5. Discussion:

In summary, the CNN classifier performed as expected, with respect to the comparative performance of other conventional classifiers (83% vs the maximum 68% shows a remarkable improvement). This agrees with the findings of comparable literature such as Helber et al. (2019) which noted that the recent use of the state-of-the-art convolutional neural networks (CNN) is able to achieve superior results in image classification than other classifiers, and how deep CNNs performed better than shallow CNNs. Therefore, we were able to meet our goal of comparing a series of machine learning algorithms to identify the best classifier to further our remote sensing applications (e.g., classifying land uses in a selected random European region).

Having said that, a possible reason that the model has not performed even higher than 83% might be due to the relabeling, especially the new label "Water". This is because the "River" images also contain a fair amount of land area, making it hard to distinguish from other categories. "Forest" achieved high predictive accuracies across all models mainly because it was very distinct (i.e., green thickets) and was not merged with the other original classes.

Reference:

[1] Helber, P., Bischke, B., Dengel, A., & Borth, D. (2019). Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *12*(7), 2217-2226.

[2] Helber, P., Bischke, B., Dengel, A., & Borth, D. (2018, July). Introducing eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. In *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium* (pp. 204-207). IEEE.

[3] Di Gregorio, A. (2005). *Land cover classification system: classification concepts and user manual: LCCS* (Vol. 2). Food & Agriculture Org..

[4] Talukdar, S., Singha, P., Mahato, S., Pal, S., Liou, Y. A., & Rahman, A. (2020). Land-use land-cover classification by machine learning classifiers for satellite observations—A review. *Remote Sensing*, *12*(7), 1135.

[5] Masuch, L (2016, December 6). Deep Learning - A Visual Introduction. Retrieved from https://www.slideshare.net/LuMa921/deep-learning-a-visual-introduction.

[6] Chengwei (2019). Simple Guide on How to Generate ROC Plot for Keras Classifier. *Dlology*. Retrieved from https://www.dlology.com/blog/simple-guide-on-how-to-generate-roc-plot-for-keras-classifier/.

Land Cover Classification and Identification_Final Report
MUSA-650- Machine Learning in Remote Sensing

Label Reference (0 to 5):

	label	count	range_end	description
0	0	2400	2400	Agriculture
1	1	2400	4800	Forest
2	2	2400	7200	HerbaceousVegetation
3	3	2400	9600	Highway
4	4	2400	12000	Building
5	5	2400	14400	Water