APPLIED MACHINE LEARNING SYSTEMS II (ELEC0135 20/21) REPORT

SN: 19162034

ABSTRACT

This paper explores the best approach to solve a classification problem with 6 classes. The goal will be to implement a new architecture that could better the results obtained in the original Kaggle competition. An overview of models considered is given, choosing VGG as the one that offers best results while also considering resources used. A model of experts architecture is implemented to combat the high confusion between certain classes. Two experts are trained, and their results combined with a base classifier to predict the class. An ablation study will prove that the model without using the base classifier has similar results while using much fewer resources. The results find that there is a high degree of confusion between two of the classes, which seem to have a correlation that is too high for the algorithm to distinguish with a high degree of certainty. However, the final results are satisfactory, with a higher accuracy score than the original competition participants, with all models trained in as few epochs as possible.
GitHub link:
https://github.com/sebbarpar/AMLSII_assignment
Competition link:
https://www.kaggle.com/puneet6060/intel-image-classification

1. INTRODUCTION

Classification problems can prove challenging due to the incapability of the model to distinguish between certain classes. This confusion can be combated, but often adding more epochs simply increases overfitting. For this reason, new models can be trained which will be dedicated to some of the total classes. This way the problem is segmented, and if the model is not confused when dealing with more generic classes, the accuracy can improve. This is the basic thinking behind this project. However, this is a basic architecture and the specifics have to be explored.
The first phase of the project will be choosing a CNN architecture. Three of them will be considered, which will be tested on the base classifier, which is to distinguish between 6 classes. The CNN architecture with the best performance will be used for the rest of the project. Next, the classes with the highest confusion will be assigned a dedicated CNN to distinguish between them. Apart from this, an additional CNN will be implemented to determine if the class of the image belongs to one of these. The architecture can be found more clearly in Figure 5. Lastly, the overall architecture will be tested and an ablation study performed to try and decrease the resources used.

2. LITERATURE SURVEY

The models considered for this task were mainly convolutional neural networks, since it is the most common way to approach image classification tasks with this many classes. Other approaches were considered such as extracting features and using other algorithms, such as k-nearest neighbors [1], but this was discarded due to the disparity of the images. It would be very challenging to extract comparable features for such different images. The images could be reduced to feature vectors and later apply simple machine learning algorithms on them, but the usage of CNNs was considered more sophisticated and worthwhile [2].
Three different architectures were considered:

1. VGG [3]: Introduced by Simonyan and Zisserman in 2014, this is a very simple but flexible network, since the number of convolutional layers can be increased or decreased easily, as they are stacked on top of each other. Max pooling layers are used to reduce the size of the images.
2. Inception [4]: Introduced by Szegedy in 2014, it is considered a micro-architecture, since many of them can be stacked on top of one another. It is based on extracting features at multiple levels, since it computes convolutions of different sizes on the same module of the network.
3. ResNet [5]: Introduced by He, Zhang, Ren and Sun in 2015, it is based on a module that skips some steps in the model. It stacks a convolutional layer from the input and then the output is connected to the original input in order to pass information from earlier era to later ones.

Once the best architecture is chosen, a mixture of experts [6] model will be tested. One of the oldest and most used combining methods in machine learning, where the problem is divided into easy-to-manage subsets that will be treated independently. What this means for the project in hand is that different modules will be implemented to distinguish between different classes, bringing it together to determine the overall class.

3. DESCRIPTION OF MODELS

CNN networks are used to predict the classes of the images. This section will also analyze how the approach differs from the competition entries. There are three phases to the project:

3.1. Initial classifier.

In this part, a CNN network is trained to predict the class of the image. There are 6 classes in total: buildings, forest, glacier, mountain and street. Three different architectures will be tested. Two metrics will be used to evaluate each of the algorithms: speed and accuracy. All models were trained in equal circumstances, so while the times of execution might be quite high, they would decrease in a more powerful machine. However, the relation between them should remain the same. Accuracy will be measured for train and validation set to account for overfitting.

3.1.1. VGG:

The VGG has a very simple architecture primarily composed of convolutional layers stacked on top of each other, and reducing the size by max pooling. There is a degree of liberty when implementing this architecture, since the number of layers that can be stacked is up to the user. For this project, two variations have been compared:

A. The second model has three sets of two-dimensional convolutional layer, which make the image 18x18 after the last layer. This is then flattened and densed to fit the number of classes being evaluated. Figure 1 shows the architecture.

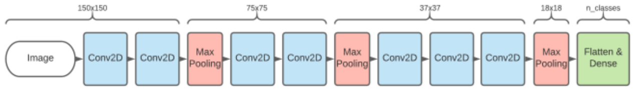


Figure 1: Large VGG architecture

B. The second model is smaller, with one less convolutional layer to decrease computation time. This can be considered an ablation study to check if reducing the number of convolutional layers will greatly impact the performance. Figure 2 shows the architecture of this model.

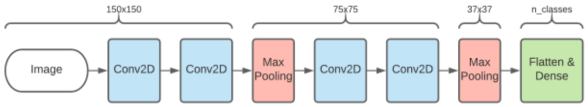


Figure 2: Small VGG architecture

3.1.2. Inception:

The inception network does not work as sequentially as the VGG. It performs 1x1, 3x3 and 5x5 convolutions on the same module of the network. These are later concatenated (along with a max-pooling layer). The architecture can be found in Figure 3.

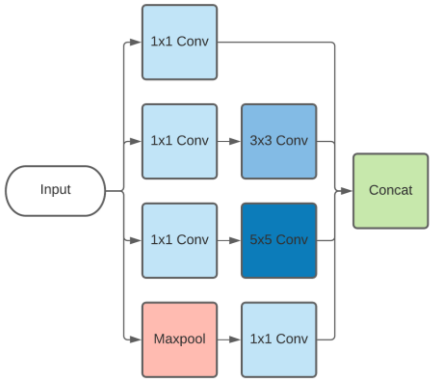


Figure 3: Inception architecture

After concatenating, the output can be fed back into the model. However, this produces a model that is too big for our purposes, so this will be discarded. Therefore, after this the layers to make a classification problem are added (flatten and dense).

Another version was tested, eliminating the 1x1 convolutional layers in the second, third and fourth row of the diagram to try and decrease the size of the model. The accuracy was terribly low and the resources saved were minimal, so this method was discarded. The results were so much inferior that it was not included in section 4.

3.1.3. Residual

Flatten and dense were added to the original architecture again to make it applicable to a classification task. Figure 4 shows the architecture.

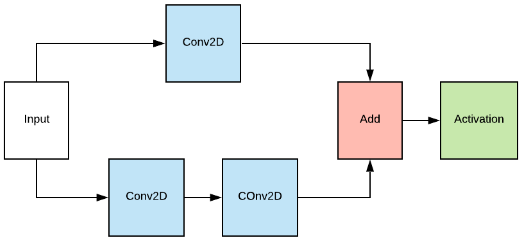


Figure 4: Residual architecture

3.2. Mixture of experts.

In order to better the results obtained with the best performing model, a mixture of experts architecture is implemented. What this does is detect which classes have a high confusion coefficient between them and trains a dedicated model to distinguish them. Based on results obtained from the confusion matrix that will be seen in the next section, two experts need to be trained.

3.2.1. Classifier 1.

Named man-made classifier from now on, it will distinguish between forest, building and street classes. Building and street have an especially high correlation, while forest is added to the expert classifier as these are the only two classes where the confusion is significant.

3.2.2. Classifier 2.

From now on called nature classifier, this expert will distinguish between sea, glacier and mountain.

3.3. Final model

This model will encompass all the previous ones. The structure can be seen in Figure 5.

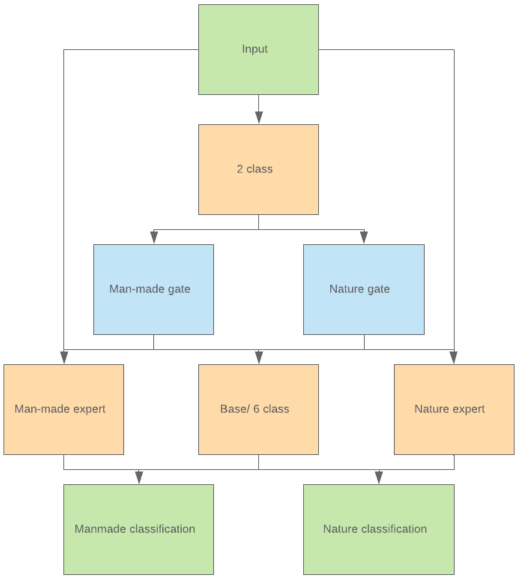


Figure 5: Mixture of experts architecture

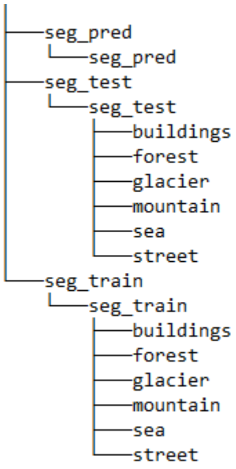
The orange modules are the ones pretrained to build the mixture of experts, the blue ones are the gates that will be trained and the green ones represent input and output. The first classifier will decide if the image belongs to the man-made classifier or to the nature classifier. The gates will decide which expert will be used, along with the 6-class classifier. The final results will be calculated as the sum of the multiplication of the output for every classifier and its importance.

3.4. Differences from competition

A common theme in the competition solutions is using only one CNN architecture and training it for many epochs. At times this architecture ends up being very big, to the point that when trying to recreate their results, the time it would take is too high. Therefore, a more sophisticated approach was taken to reduce the number of epochs needed to achieve good results. A MoE model will reduce the time needed to train the model and is expected to achieve decent results. When using a different and larger architecture, the validation scores are very high, reaching the 90s in accuracy. The goal of the project is not to imitate these results or try to improve on them, but to utilize less resources and still achieve a decent accuracy score. Other solutions proposed multiple ensemble CNNs, which again was considered too many resources utilized.
Apart from this, many competition entries used preconfigured models from keras, while in this project the CNN architecture was implemented by hand.

4. IMPLEMENTATION

The dataset is composed of almost 25000 RGB 150x150 images, separated in three folders: train, test and pred. The purpose of this last folder is to test the model on a new set of images, but labels for these are not included. Therefore, the images in this folder will not be used, since scores won't be able to be determined, and the performance of the model can't be assessed. The other two folders have the images organized as seen in Figure 6.



```
├───seg_pred
│   └───seg_pred
├───seg_test
│   └───seg_test
│       ├───buildings
│       ├───forest
│       ├───glacier
│       ├───mountain
│       ├───sea
│       └───street
└───seg_train
    └───seg_train
        ├───buildings
        ├───forest
        ├───glacier
        ├───mountain
        ├───sea
        └───street
```

Figure 6: Dataset structure

To assign labels to each picture, the name of the folder was checked, and if the picture was inside the "buildings", the label was 0. If it was inside the "forest" folder, it was 1… and so on. Since there are 6 possible classes, the labels are 0-5.
To try and deviate from the competition format, a new train/validation/test split was made. The images from both test and train folders were mixed, shuffled and split again in three sets: train, validation and test. The train set will be used to train the model, the validation to assess its initial performance and tune the parameters, and the test to provide an untouched dataset to test the model on after the validation phase.
The main libraries used for this project were Keras, cv2 and numpy.
As explained before, the project was separated in phases. The implementation of each phase can be found in the corresponding section.

4.1. Initial classifier.

Different architectures are compared to see their accuracy and performance on the 6-class classifier. As explained before, 3 algorithms (and one variation of an algorithm) will be tested and the best one will be the chosen one to continue with the project. Since it would be an arduous process to have to compare all these algorithms for every step in the development of the final model, the results from this part will be used to justify which algorithm will be used for the rest of the project.

4.1.1 VGG

A. Large VGG

A VGG model was tested, with three convolutional layers. The results are satisfactory, but the training takes a long time, and it would be interesting to see what would happen if one convolutional layer was taken out.

The final scores can be found in Figure 7.

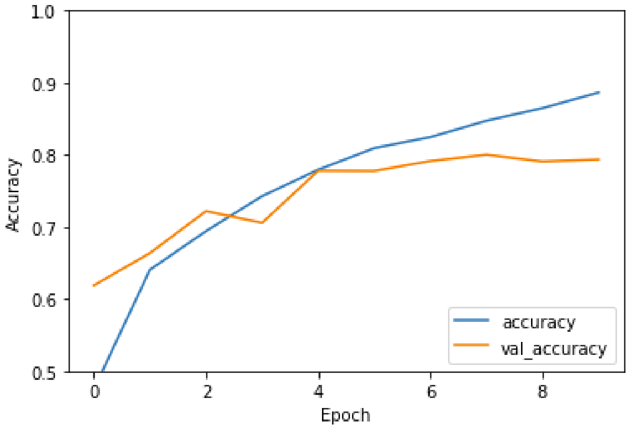


Figure 7: Large VGG accuracy

B. Simple VGG

A CNN following the VGG model was built, with two convolutional layers. The number of epochs was set to 10. Figure 8 shows the scores for training and validation sets according to each epoch.

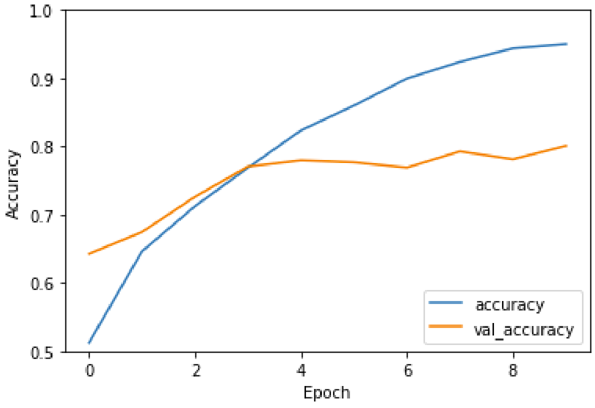


Figure 8: Simple VGG accuracy

As it can be seen, after the third epoch there is overfitting to the training dataset. This is in line with findings in the competition, where the a team achieved a train accuracy of 96% and validation of 82%. The final scores for the model were 95% for training and 80% for validation, but with only 10 epochs vs 35 epochs for the other model. Since after the third epoch there is overfitting, the model chosen to work with was the one after three epochs. This model performs on par with the large VGG, but utilizes fewer resources, so it will be the chosen algorithm.

4.1.2. Inception

The Inception algorithm was tested on the dataset. The algorithm was obtained from (), but a flatten, dropout and dense layers had to be added to adequate it to a classification problem with 6 classes.

The results after 10 epochs can be found in Figure 9.

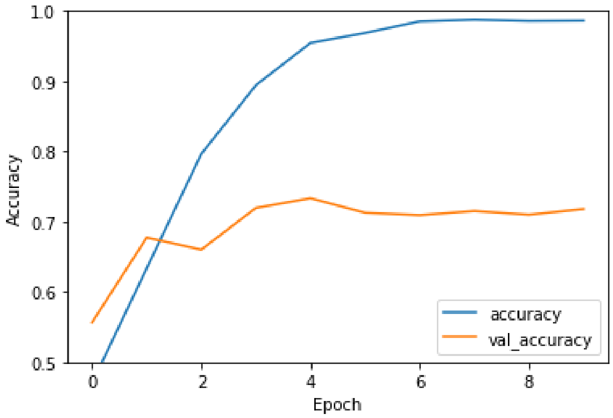


Figure 9: Inception accuracy

After the third epoch there is overfitting, same as in the VGG model. However, the accuracy is lower and the model is more computationally complex. The inception model took 4 times more per step in every epoch when compared to the simple VGG model. Since it takes longer than the VGG and produces worse results, this algorithm will not be used in the future.

4.1.3. Residual

The residual architecture was tested on the dataset. Again, flatten, dropout and dense layer had to be added. The results can be found in Figure 10.

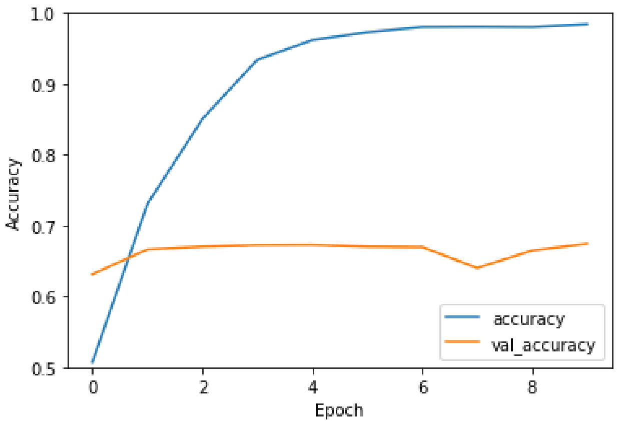


Figure 10: Residual accuracy

There is overfitting after just one epoch, but the accuracy results are the lowest of any architecture. The execution speed is faster than the inception architecture, but still more than VGG (1.5x). For both these reasons, this architecture will also not be implemented again.

4.2. Mixture of experts

The simple VGG architecture provides reasonable results and could be used to predict the classes satisfactory. However, improvements can be made to try and increase the score. A mixture of experts architecture is implemented to improve the performance of the model. In order to do this, the number of images being classified incorrectly must be studied. For example, if many images of class 1 are being classified as class 2 and vice versa, the expert could be implemented to distinguish these two classes and therefore get better results. Figure 11 shows the confusion matrix for this base classifier.

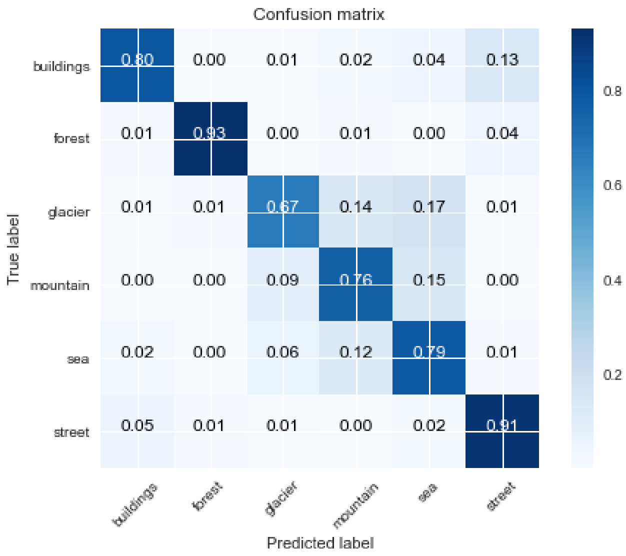


Figure 11: Base classifier confusion matrix

As it can be seen, the label predicted with least accuracy is the glacier, while the forest is the one with most. However, there is some confusion between buildings and street, something quite reasonable, as well as with these two classes and forest. There is also confusion between glacier, mountain and sea. Therefore, two experts will be implemented: one to distinguish buildings, street and forest and another to distinguish between glacier, mountain and sea.

4.2.1. Two-class classifier.

The same CNN from 4.1 was used to design a classifier that now only distinguishes between two classes:

1. Buildings, street or forest.
2. Glacier, sea or mountain.

This classifier was also trained for 10 epochs to find out when the overfitting commences. The results can be seen in Figure 12.

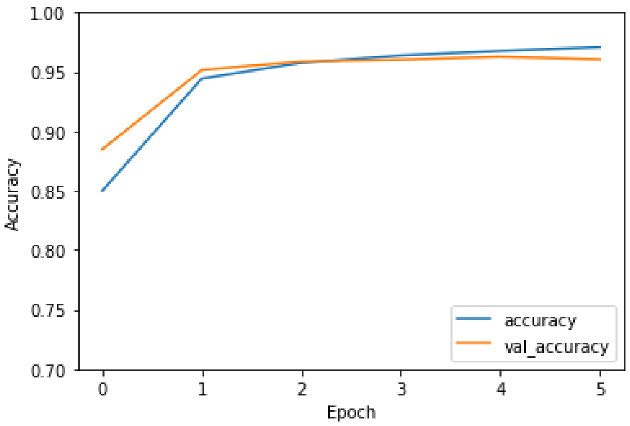


Figure 12: Two-class classifier accuracy

Very good results are obtained after only one epoch, and therefore this is the number of epochs the model will have. Figure 13 shows the confusion matrix for this classifier.

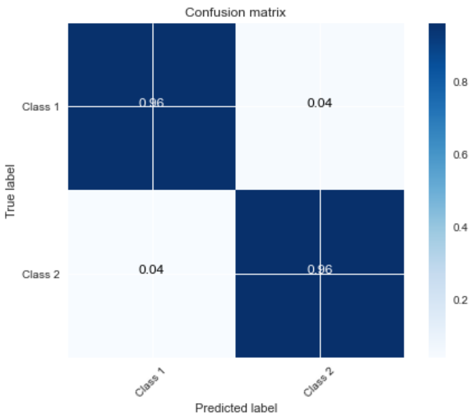


Figure 13: Two-class confusion matrix

The results are very satisfactory, with very high accuracy for both classes. This is reasonable, as the confusion matrix for the original 6-class classifier shows little confusion between the labels inside these two classes.

The next step will be to train another two classifiers:

1. To distinguish between building, street or forest. This classifier will from now on will be referred to as "man-made".
2. To distinguish between glacier, sea or mountain. This classifier will be called "nature".

In order to do this, before the train, validation and test splits the dataset is modified to include only images that belong to one of these labels.

4.2.2 Man-made classifier.

The dataset is altered to only have in it pictures belonging to three classes: street, building and forest. The model is then trained with these images and 6 labels, but obviously it will only be able to predict for these three. Again, the VGG architecture was the one used. The model was trained for 10 epochs and the results can be found in Figure 14.

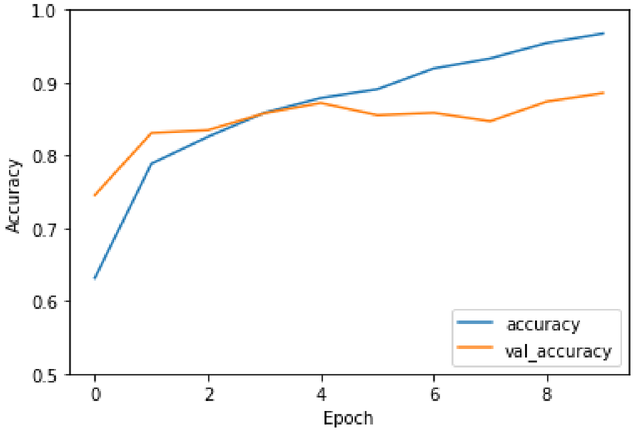


Figure 14: Man-made accuracy

As it can be seen, the results after just one epoch are very satisfactory, but overfitting does not begin until after the second or third epoch. Therefore, three epochs will be the chosen amount for this algorithm. The confusion matrix can be found in Figure 15.

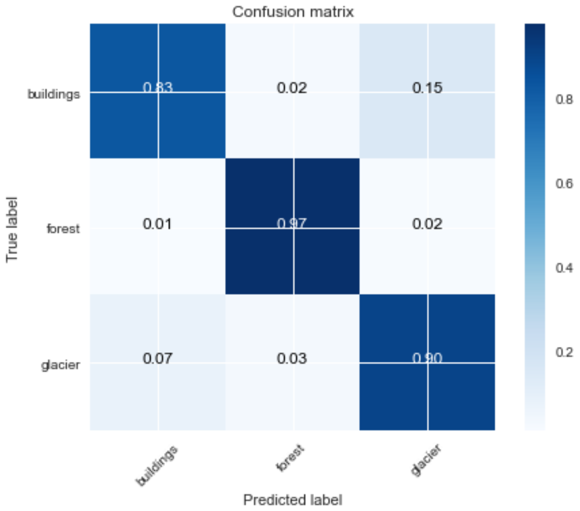


Figure 15: Man-made confusion matrix

Although the street class sees a slight decrease in accuracy, buildings and forest improve.

4.2.3. Nature classifier.

The nature classifier is built to distinguish between three classes. In the original 6 class classifier these three had the highest confusion coefficients between them, and therefore it is expected that this classifier will need more epochs to be trained before there is a gross amount of overfitting. The accuracy results for each epoch are shown in Figure 16.

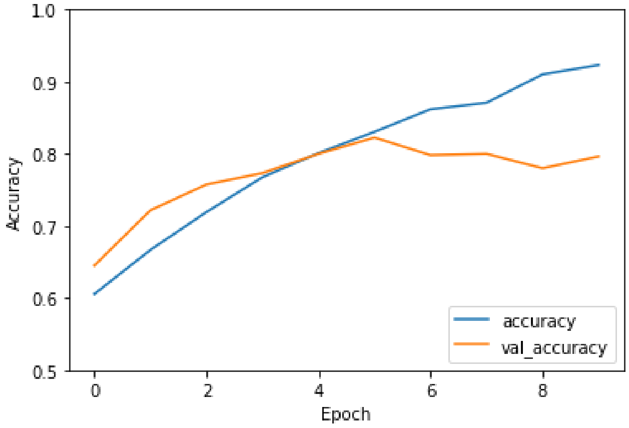


Figure 16: Nature accuracy

Overfitting begins after epoch 5, so this is the number of epochs that will be used. The confusion matrix is shown in Figure 17.

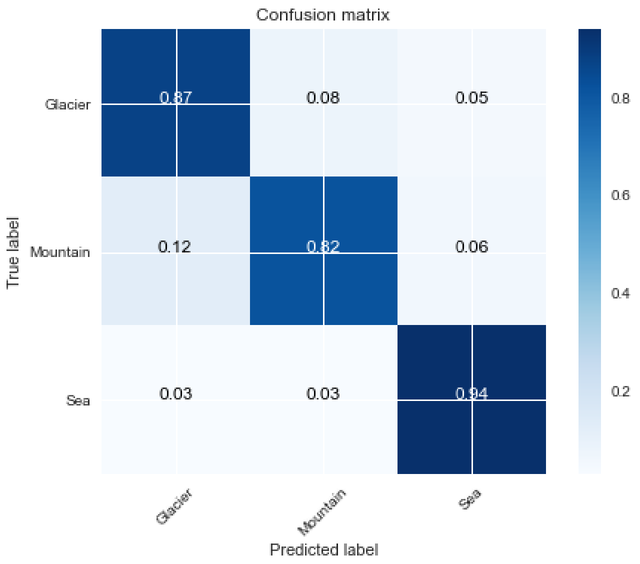


Figure 17: Nature confusion matrix

Comparing these results with the confusion matrix of Figure (), it is clear that there is an improvement. This will be further discussed in Section 5.

4.3. Final architecture.

For the final part of the project, all models have been trained. Two new gates are created and have to be trained. These are used to assign weights to the expert model and the 6-class classifier. The results after three epochs of training can be found in Figure 18.

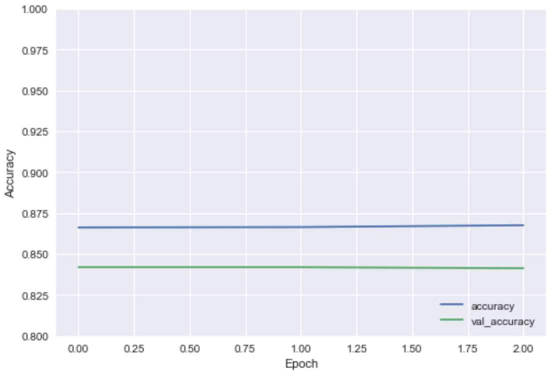


Figure 18: Final model accuracy

Figure 19 shows the confusion matrix for the final model.

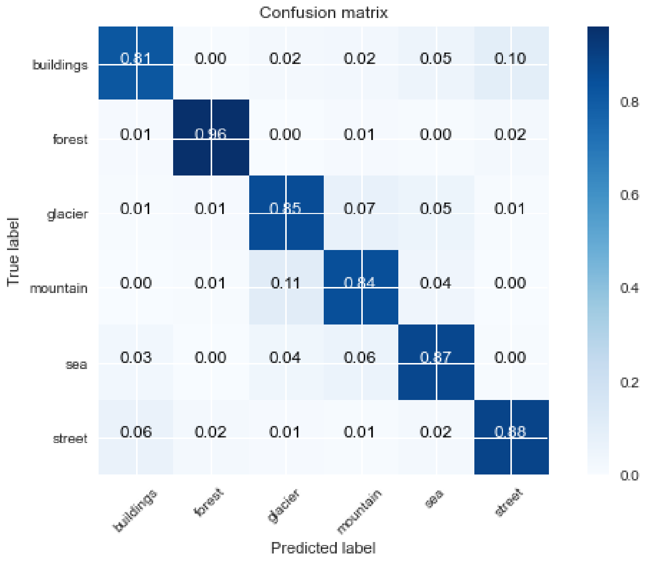


Figure 19: Final model confusion matrix

When comparing with the matrix obtained by the base classifier, all accuracy scores have increased except for the street class.

4.4. Ablation studies.

After the model achieved such good results, an ablation study is in order to test if it could achieve similar results using less resources. The first ablation study that could be considered to have happened in the project is choosing a smaller version of the VGG architecture instead of the large one. This section however will concentrate on changes to the final architecture. Three studies were made:

4.4.1. Remove gates

The results for validation and train accuracy are suspicious. Such good results after one epoch and specially without training the subgates must mean that they do not need training, and therefore they are probably not necessary. Therefore, an ablation study was conducted where a second model was compiled without these two extra gates. With this, since all weights were frozen before, only the last lambda gate is trainable, but all its values depend on weights from other algorithms. The confusion matrix can be seen in Figure 20.

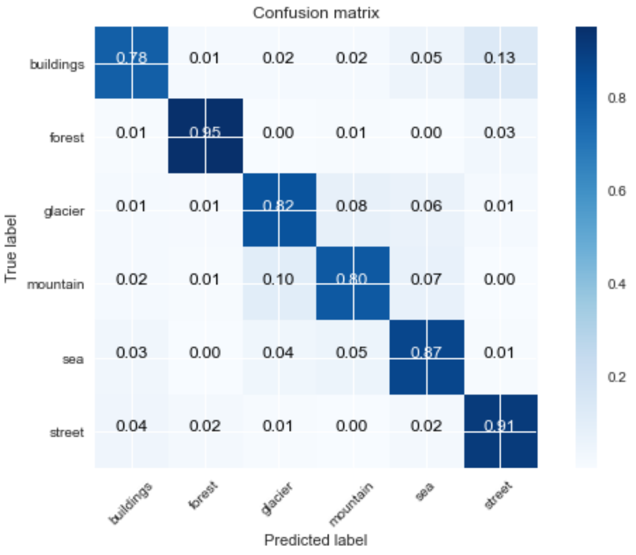


Figure 20: Final model without gates confusion matrix

As it can be seen, the results are slightly worse, especially for the buildings class. This is in line with what was seen in the previous section. The way the final result is predicted is by adding the contribution of the base expert to each class and that of the corresponding expert. Since the experts obtain better results than the baseline when evaluated, giving more weighting to the experts might improve the accuracy. They are weighted with double the importance of that of the base classifier, and the confusion matrix is seen in Figure 21.

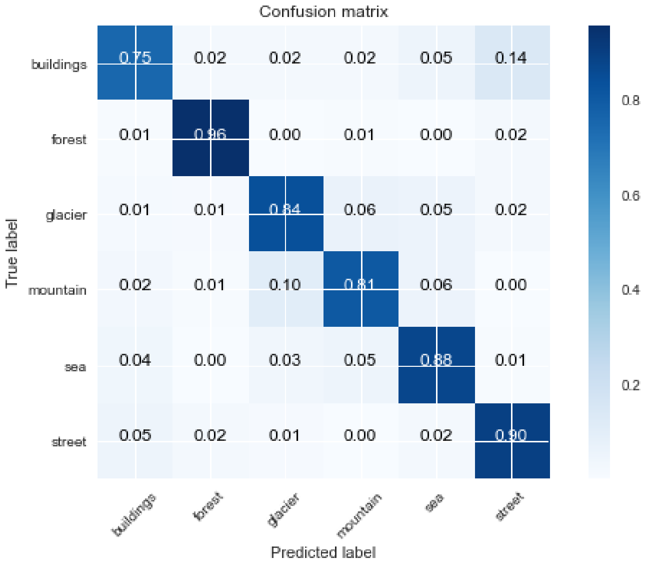


Figure 21: Final model with more weight for experts confusion matrix

The results are very similar, and in some cases better. This observation motivates the next part of the ablation study. If it does not make a big difference whether the base classifier is weighted as 1 or ½, this classifier must not be very important to the overall model, as will be tested in the next section.

4.4.2. Remove base classifier.

The main architecture of the model consists of four CNNs: base classifier, gate and two experts. Taking out the base classifier might be a bold decision, but if the results are not heavily impacted, the model's size will be decreased by a great amount. The confusion matrix for this model can be seen in Figure 22.

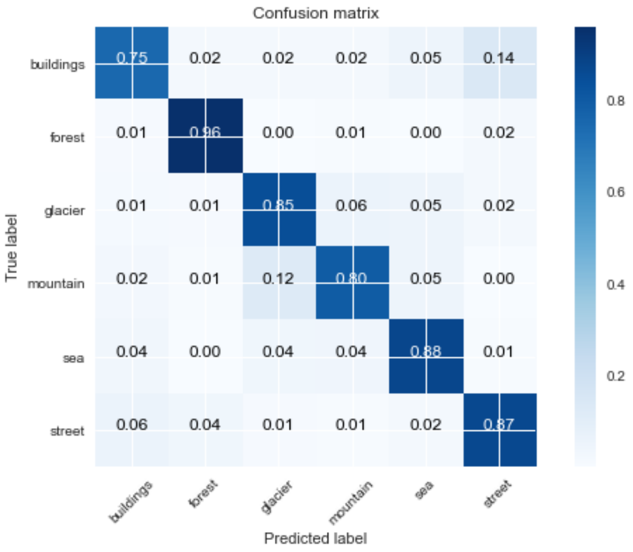


Figure 22: Final model without base classifier confusion matrix

The results of the ablation studies will be analyzed further in section 5.

4.4.3. Remove two-class classifier

A final step to the ablation study, removing the two-class classifier and adding the contributions of nature and man-made experts for every image. This would greatly reduce the size of the model, but it might offer poor results since there is no gating. Figure 23 shows the confusion matrix.

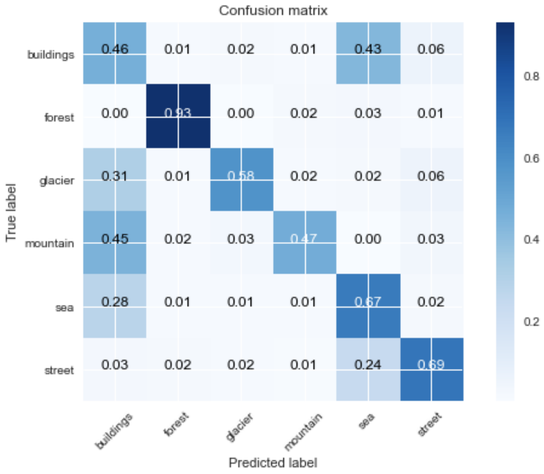


Figure 23: Final model without two-class confusion matrix

Clearly, the results are very poor, as there is a high degree of confusion between nature and man-made classes. Therefore, the two-class classifier is vital to the model.

5. EXPERIMENTAL RESULTS AND ANALYSIS

Since results from each section were driving the way the project would progress, some of them have already been mentioned. In this section, however, they will be studied more deeply and explained.

5.1. Initial classifier.

Table 1 shows the training, validation and test accuracy for each of the three algorithms considered for performing the 6-class classifier.

Algorithm	Train	Validation	Test
VGG	80.1	80.9	80.6
VGG +	80.5	80.8	80.9
Inception	74.2	75.6	74.6
Residual	78	77	77

Table 1: Accuracy for base classifiers

As explained in section 4.1, accuracy is not the only significant metric. The time the model takes to train was also taken into account. The large VGG model offer slightly more accuracy than the simple VGG, but every step of the training takes 3x as much time. For this reason, it is not reasonable to use this algorithm instead of one that offers slightly worse results. The Inception and Residual had lower accuracy scores, and therefore were discarded.

5.2. Mixture of experts

Table 2 shows the accuracy scores of each expert and the two-class classifier.

Module	Train	Validation	Test
Nature	86.5	85.9	87.4
Man-made	90	88.9	89.9
Two-class	95.4	95.6	95.8

Table 2: Accuracy for MoE modules

All scores are much higher than the base classifier, which is expected as there were fewer classes to compare, but the most important is the two-class classifier, since this will determine which path the images will follow in the final architecture. If an image is mislabeled as nature when it belongs to man-made it is very unlikely that it will eventually be classified correctly, since the importance of the base classifier must be much higher than that of the nature expert. Therefore, this is the truly important score, which is very adequate.

5.3. Final architecture

Table 3 shows the scores of the final model architecture.

	Train	Validation	Test
Final	87	85	86.8

Table 3: Accuracy for final architecture

Comparing these scores and the confusion matrices of the final architecture vs the initial base classifier, there is no doubt that significant improvements were made, especially in the classes belonging to the nature classifier. However, some further comments can be made. Figure 24 shows the confusion matrix of the final model and the nature and man-made matrices.

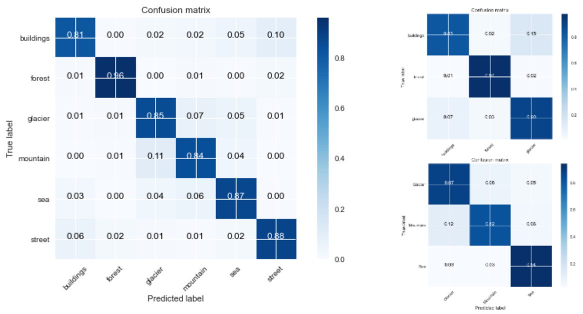


Figure 24: Confusion matrices of final model and experts

A couple observations can be made:

- All accuracy scores are lower for the final model, which makes sense considering that the two-class classifier does not have perfect accuracy, and some might be getting mislabeled as nature or man-made.
- There is less confusion between buildings and street in the final model. This means the baseline classifier is helping distinguish between these two classes.

5.4. Ablation studies

Table 4 shows the scores of the models after the ablation studies.

Variation	Train	Validation	Test
No gates	85.3	85.1	85.6

No base	84.6	85	85
No two-class	63	64	63

Table 4: Accuracy for ablation studies

As explained before, it was unexpected how the accuracy for the dataset was so high even before training the gates. With the gates taken out and simply adding the contribution of base model and experts, the results are very similar, while slightly lower. Taking the base expert out reduces the accuracy by another half percent. When removing the two-class classifier however, the results decrease massively, and therefore the two-class gate will remain part of the model. The ablation study was a success, since the utilized resources are much fewer than before but the accuracy was barely affected. Some observations can be made when comparing the matrices.

- The buildings class is the most affected by the ablation. As said before, this leads to believe that the base classifier is helping distinguish it from other classes, and if this is taken out, the accuracy will decrease.
- The accuracy scores decrease mainly because the two-class classifier now does not have any "help" from other classifiers to correct its errors. The results from the final algorithm with no base can be calculated by:

$$\frac{\overline{acc(nature)} + \overline{acc(manmade)}}{2} \times \overline{acc(twoclass)}$$
$$= \frac{86.6 + 89.6}{2} \times 95.6 = 84.5$$

It is worth mentioning that the accuracy score of the buildings class was attempted to be improved by increasing its weight, but it was seen that when its score improved, the street class fell by the same amount. Clearly, the accuracy of these two classes is a trade-off, since they are very similar. Figure 25 shows an image of buildings and another of street.

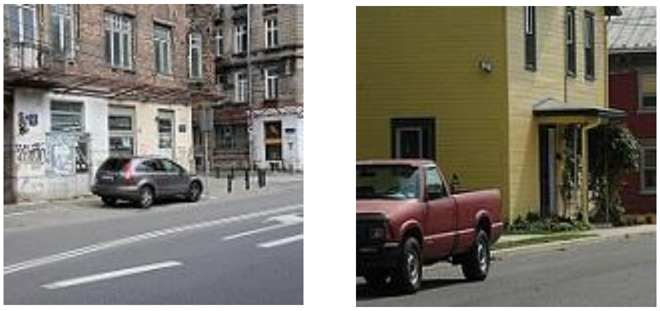


Figure 25: Building(left), street (right) images

Can the reader tell which is which? It stands to reason that the machine might classify these as the same class, and therefore for these classes to not be independent.

5.5. Extra images

When separating train, validation and test, the test dataset was not used when training the model, as to have some images that have been untouched on which to test the algorithm to account for overfitting. However, all these images belong to a precompiled dataset. In order to be sure of the functionality of the algorithm, some extra images were chosen at random and the algorithm tried to predict their class. Figure 26 shows 6 images, one from each class, and their predicted class on top.

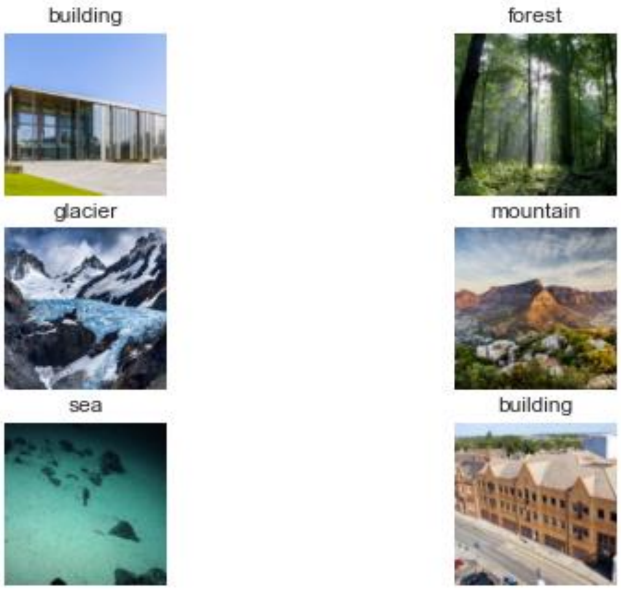


Figure 26: Test images and predicted classes

As it can be seen, it predicts the class of the image well for most of them, but the last image is supposed to be "street" class, but it is classified as building. As explained before, these are the classes with the highest confusion between them, and therefore this is a reasonable mistake.

6. CONCLUSION

The chosen architecture has provided very satisfactory results, with scores that are lower than those in the competition but using much fewer resources. The ablation study specially was very useful to reduce the size of the model and improve upon the competition results.

7. REFERENCES

[1] Cunningham, Padraig & Delany, Sarah. (2007). k-Nearest neighbour classifiers. Mult Classif Syst

[2] I. Kanellopoulos & G. G. Wilkinson (1997) Strategies and best practice for neural network image classification, International Journal of Remote Sensing, 18:4, 711-725, DOI: 10.1080/014311697218719

[3] Simonyan, Karen & Zisserman, Andrew. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv 1409.1556.

[4] C. Szegedy et al., "Going deeper with convolutions," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 2015, pp. 1-9, doi: 10.1109/CVPR.2015.7298594.

[5] He, Kaiming & Zhang, Xiangyu & Ren, Shaoqing & Sun, Jian. (2016). Deep Residual Learning for Image Recognition. 770-778. 10.1109/CVPR.2016.90.

[6] Masoudnia, Saeed & Ebrahimpour, Reza. (2014). Mixture of experts: A literature survey. Artificial Intelligence Review. 42. 10.1007/s10462-012-9338-y.