Introduction

Land cover detection is the process of analysing aerial imagery to identify and map the different types of ground structures in a given area, for example forests, buildings or rivers.
It is critical for a variety of applications, including environmental monitoring, urban planning, agriculture or natural resource management.
Manually evaluating the images is rather time consuming and costly.
Therefore, the usage of machine learning does not only provide a faster, but also scalable solution to this problem.

As it is rather challenging to correctly classify 21 (partly similar) classes, this paper introduces two different approaches to implement the automated detection of different structures.
The first model is a Convolutional Neural Network (CNN), which is part of the sub-area of deep learning. CNNs have been developed specifically for image processing and have proven to be extremely effective for image recognition. [3][4] They are able to extract complex features in images and recognise hierarchical patterns.
The second model is a Support Vector Machine (SVM). A Support Vector Machine divides a set of objects into classes in such a way that the widest possible area around the class boundaries remains free of objects. [6]

The underlying dataset is the UC Merced Land Use Dataset [1]

Method

Data Preprocessing

The image dataset provides two different resolutions; low resolution: 61x61 pixels, and high resolution: 242x242 pixels (not used here), both with 2,100 samples. Each class contains 100 images.

The dataset was preprocessed to ensure better generalisation and model stability.
This includes the following steps:

1) Normalising and augmenting the data (afterwards 50% original data, 50% noisy data) to improve the real world performance when many different scenarios occur [3]
   This was achieved by varying the images (rotating, shifting, zooming, flipping)
2) Splitting and shuffling the data into training (80%), validation (10%) and testing (10%) to train and later evaluate the model
3) Separately for the SVM: converting the image data with Histogram of Oriented Gradients (HOG) to extract the image features as the SVM needs a clear dividing line between classes

Models

1. Convolutional Neural Network

Feature Extraction

CNNs use small neural networks (convolutional filters) that glide over the image and generate activations based on local patterns. [2][4]

The model starts with a convolutional layer with 32 filters of size (3, 3). This layer applies convolutional filters to the input image, detecting characteristics such as edges, textures, and colours.
Following each convolutional layer, a MaxPooling2D layer with a pool size of (2, 2) is applied. MaxPooling reduces the spatial dimensions of the feature maps, retaining the most important information while discarding less relevant details. This helps in reducing the computational burden and prevents overfitting by focusing on the most significant features. [2]
After the convolutional and pooling layers, a Flatten layer reshapes the output from the convolutional layers into a flat vector. This prepares the data for processing by Fully Connected layers. The Dropout layers help in regularisation by randomly dropping out a fraction of neurons during training, reducing overfitting. [4]
The final Dense layer consists of neurons equal to the number of classes in the classification task, each neuron outputting the likelihood of the certain class being correct. [4]

Feature Processing

Extracted features get passed through interconnected layers of neurons. These connections enable the network to grasp intricate non-linear correlations between features and encode abstract concepts essential for land cover classification.

Activation functions play a crucial role in CNNs. They introduce non-linearity into the network's computations, aiding in the capture of complex patterns within the data. [5]

Feature Training

The training process involves optimising the weights of the neural connections in the network to maximise classification accuracy. The model uses Categorical Cross Entropy as a loss function and Adam as a gradient descent algorithm for optimising.

Feature Testing

The CNN is applied to the images in the test set to generate the predicted land cover class for each image. The accuracy is calculated over the testing data and additionally displayed by sampling 100 random images and their predicted classes.

2. Support Vector Machine

Feature Extraction

SVMs are particularly suitable for tasks where there are clear separating surfaces between the classes. [7] Therefore, the data was first transformed with HOG (as described in *Data Preprocessing*) and then limited to five well distinguishable classes.

Feature Processing

Each training example is represented by a feature vector consisting of the extracted features. SVMs can find optimal hyperplanes in feature space that maximally separate data points of different land cover classes. These hyperplanes represent linear relationships between the features that are relevant for the differentiation of land cover types. [7]
The implemented SVM uses a Radial Basis Function (RBF) kernel. It is particularly useful for non-linear problems, as it projects the data into a higher dimensional space in which they are easier to separate. [6][7][8]

Feature Training

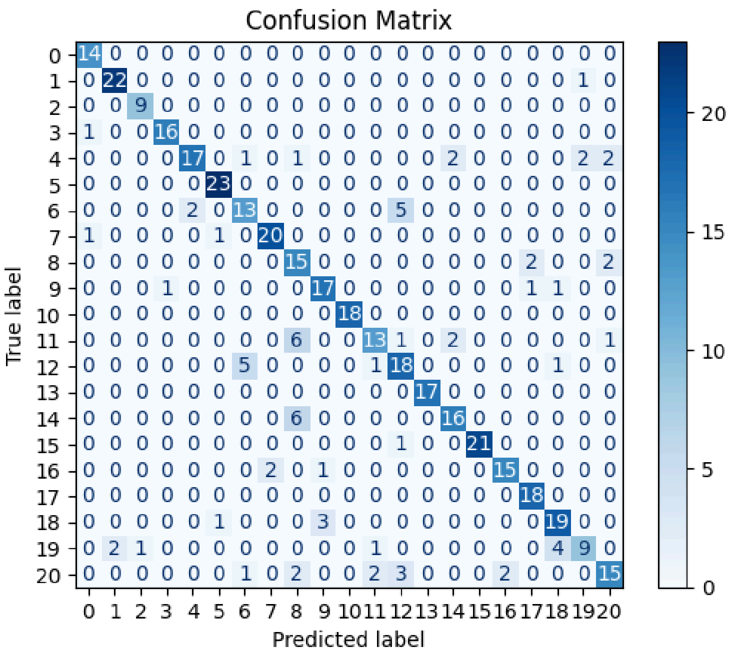
There are two important hyperparameters for the RBF-SVM that need to be optimised: Gamma and C. Gamma controls the shape of the decision boundary, while C controls the trade-off between margin maximisation and error minimisation. During training, certain training examples are identified as support vectors that are close enough to the decision boundary to define the interface. These support vectors are the ones that are used to calculate the decision function. [8]

Feature Testing

The SVM is applied to the images in the test set to generate the predicted land cover class for each image. The accuracy is calculated over the testing data and additionally displayed by sampling 100 random images and their predicted classes.

Results

The following is displayed in the results of the models:
The CNN with 21 classes reaches an accuracy of approximately 82 per cent.
The CNN and the SVM with 5 classes have an accuracy of approximately 92 and 81 per cent respectively.
The evaluation of the CNN and the SVM is based on test and validation accuracy (#correct_pred/#num_labels). As the test and validation data is completely separated from the training data, this measurement indicates the quality of the model.
The CNN also has two supporting graphs. If the validation loss curve on the loss graph begins to rise after a certain point, the model tends to overfit. [3]
A possible prevention for overfitting is early stopping, adjusting the number of epochs or the batch size. [3]



21-class matrix of the 3 provided matrices.
An ideal confusion matrix would align all numbers in the diagonal as it predicts all labels correctly.
It shows the True Positives (diagonals) and the False Positives and False Negatives (non-diagonal)

Conclusion

5 classes

Working with just five different classes severely improved the models' quality as it is easier to extract features.

21 classes (just for the CNN)

Among the 21 land cover classes, there are several rather similar classes, so it is difficult to determine all of them correctly.
Although CNNs are very effective for image detection, they are dependent on the quality of the training data. The limited training data and the low resolution make it difficult to recognise all classes accurately. As CNNs are very computationally intensive, the use of the high resolution images is not necessarily recommended.

It is crucial to adjust the different parameters such as i) augmentation, ii) network structure, iii) batch size, iv) dropout rate all together to achieve the best result.

i) transforming the images too much could decrease the quality of the images as details get lost
ii) adding more layers to the standard CNN of Tensorflow increased stability and accuracy as increasing the number of filters lets the model capture more complex patterns.
iii) a larger batch size can help to estimate the gradient more accurately and make the training more stable
iv) adding a second dropout layer enabled a more stable model to reduce overfitting.

References

Dataset

1. http://weegee.vision.ucmerced.edu/datasets/landuse.html

Convolutional Neural Network

2. Liyao Yuan. (2021). Remote Sensing Image Classification Methods Based on CNN: Challenge and Trends. https://ieeexplore.ieee.org/document/9706946

3. Raoof Naushad, Tarunpreet Kaur, Ebrahim Ghaderpour. (2021). Deep Transfer Learning for Land Use and Land Cover Classification: A Comparative Study. https://arxiv.org/pdf/2110.02580

4. Niklas Lang. (2021). Breaking down Convolutional Neural Networks: Understanding the Magic behind Image Recognition. https://towardsdatascience.com/using-convolutional-neural-network-for-image-classification-5997bfd0ede4

5. Bernhard Kainz. (n. d.). Deep Learning – Activation Functions. https://www.doc.ic.ac.uk/~bkainz/teaching/DL/L03_DL_activation_and_losses.pdf

Support Vector Machine

6. Wikipedia. (2024). Support vector machine. https://en.wikipedia.org/wiki/Support_vector_machine
7. G. Mercier, M. Lennon. (2003). Support vector machines for hyperspectral image classification with spectral-based kernels. https://ieeexplore.ieee.org/document/1293752
8. Sahel Eskandar. (2023). Introduction to RBF SVM: A Powerful Machine Learning Algorithm for Non-Linear Data. https://medium.com/@eskandar.sahel/introduction-to-rbf-svm-a-powerful-machine-learning-algorithm-for-non-linear-data-1d1cfb55a1a

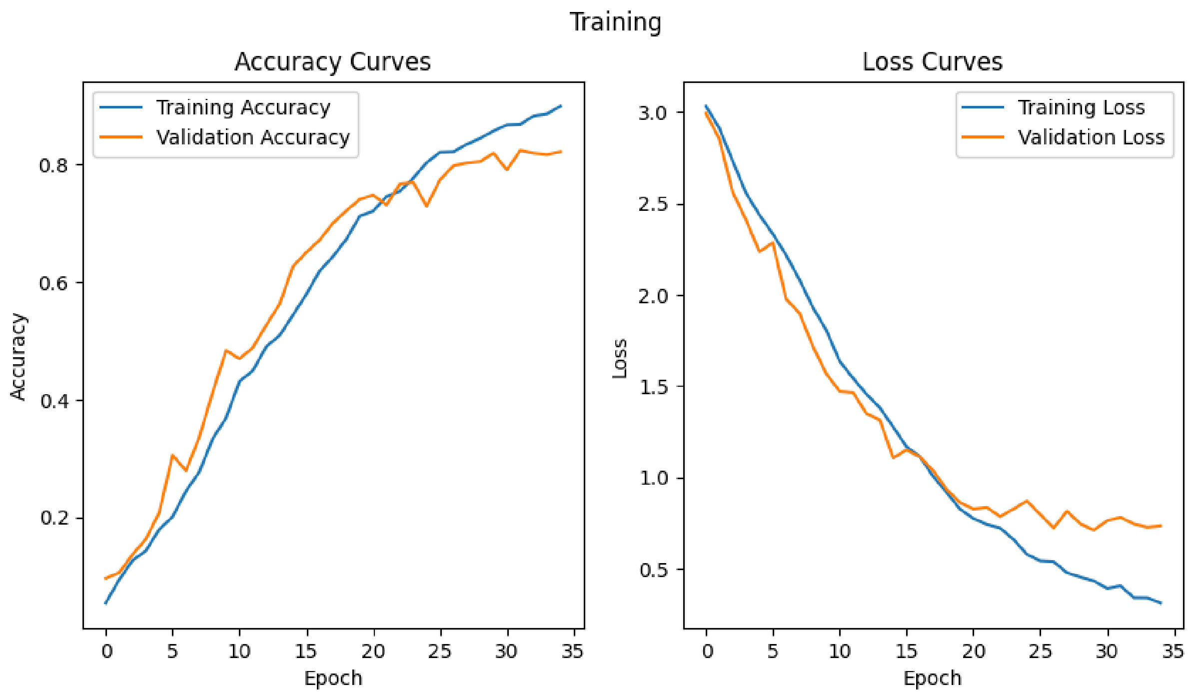
Appendices

Convolutional Neural Network

Parameters

- Augmentation Parameters (Fill Mode: nearest):
  - Width Shift Range: 0.1
  - Height Shift Range: 0.1
  - Shear Range: 0.1
  - Zoom Range: 0.1
  - Horizontal Flip: False
- Batch Size: 32
- Epochs: 35
- Dropout 0.7, 0.65

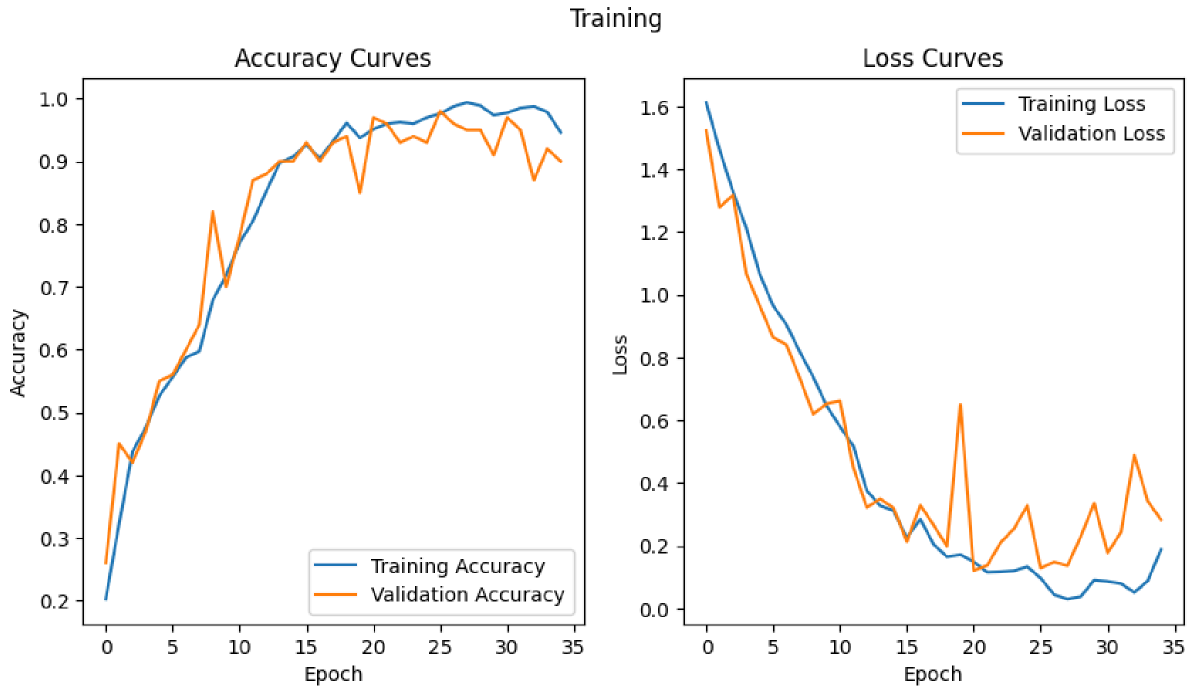
Graph for 21 classes



Training Runtime: 162.3s

Test Loss: 0.65, Test Accuracy: 0.82
Validation Loss: 0.73, Validation Accuracy: 0.82

Graph for 5 classes



Training Runtime: 41.02s

Test Loss: 0.4, Test Accuracy: 0.92
Validation Loss: 0.28, Validation Accuracy: 0.9

Support Vector Machine

Parameters

- Augmentation Parameters (Fill Mode: nearest):
  - Width Shift Range: 0.1
  - Height Shift Range: 0.1
  - Shear Range: 0.1
  - Zoom Range: 0.1
  - Horizontal Flip: False
- Feature Extraction with HOG
- Kernel: RBF

Training Runtime: 0.51s
Test Accuracy: 0.81
Validation Accuracy: 0.89