Part 1: Technical Exercice
I have prepared a concise write-up summarizing my approach and the outcomes of the model. It covers the following aspects:

Data Preparation:
The dataset consists of 105 images, comprising 43 field images and 62 road images. These images were resized to 128x128 pixels, and their pixel intensities were normalized to a range of (0, 1). The dataset was then divided into a training set (80%) and a validation set (20%).

Due to the limited number of images, data augmentation techniques were applied to the training set to prevent overfitting. These techniques included random flipping, random Gaussian blur, and random color augmentations. The aim was to enhance the diversity of the training data.

Table I illustrates the original data distribution, and Table II displays the data distribution after applying data augmentation techniques.

	Fields	Roads
Train	30	43
Val	13	19

Table I - Repartitioning of the Original Data

	Fields	Roads
Train	210	215
Val	13	19

Table II - Repartitioning of the augmented data

Model Architecture:
For this task, I employed two different models:
- The first model is an image classification encoder, which utilizes a CNN followed by a fully connected layer. The encoder module comprises 3 sets of convolutional layers with downsampling, followed by batch normalization, ReLU activation, and dropout. The final classifier includes a fully connected layer with an output size of 2 (number of classes), followed by the softmax activation function to convert the output into a probability distribution across classes.
- The second model is a multi-task neural network called "Multitask," which combines both image classification and autoencoder tasks. The model consists of an encoder, a decoder, and a classifier. The decoder architecture includes 3 transposed convolutional layers (also known as deconvolution or upsampling layers) interleaved with activation functions and normalization layers. The advantage of multitask learning lies in its ability to improve the performance and generalization of machine learning models by leveraging shared information across multiple related tasks.

Hyperparameters
During the training process, the following parameters were set:
- Batch size = 64: This choice of batch size allows for more consistent weight updates and potentially faster convergence.
- Learning rate = 0.0001: The selected learning rate is neither too large nor too small, helping to prevent divergence, oscillation, or slow convergence during training.
- Dropout rate = 0.2: By applying a dropout rate of 0.2, overfitting is mitigated, ensuring better generalization performance.
- Encoder Architecture: The encoder consists of 3 convolutional layers with 4 filters in the first layer, 8 filters in the second layer, and 16 filters in the last layer. The use of smaller filters is intended to prevent overfitting.
- Training Epochs: All models were trained for 120 epochs, ensuring sufficient iterations for convergence.

- Encoder Classification Model: The encoder classification model was trained with the cross-entropy loss function, which is commonly used for classification tasks.
- Multitask Model: The multitask model was trained with multiple loss functions. For classification, the cross-entropy loss was employed, while for the autoencoder reconstruction task, the mean squared error loss was used.

Overall, these hyperparameter choices are aimed at achieving optimal performance, preventing overfitting, and facilitating convergence during the training of the models.

Performance

The accuracy of the two models on the validation set is presented in Table III for both the original data and augmented data.

	Encoder classification	Multitask
Original data	88%	84%
Augmented data	91%	91%

Table III - The accuracy of the two models on the validation set, both with and without data augmentation.

Table III illustrates the performance of the multitask model compared to the encoder classification model. When trained on the original data, the multitask model achieved an accuracy of 88%, slightly outperforming the encoder classification model, which attained an 84% accuracy. However, interestingly, both models achieved the same accuracy of 91% when trained on the augmented data. The data augmentation has proven beneficial for both models, significantly increasing their accuracies from 88% for the encoder classification model and 84% for the multitask model on the original data, up to 91% for both on the augmented data.

In the end, considering the simplicity and good performance, I opted to choose the encoder classification model trained on the augmented data. Fig I showcases the accuracy curves of the encoder classification model trained on the augmented data.

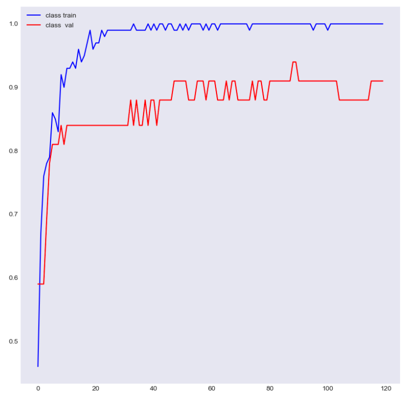

Fig 1 - Accuracy curves of the encoder classification model trained on augmented data.

The last step involves using the trained encoder classification model to make inferences on the test set, which includes 10 images (4 field images and 6 road images). The test results are presented in Table IV.

	Total	Fields	Roads
Accuracy	90%	100%	83%

I achieved an overall accuracy of 90% on the test set, with 100% accuracy for field images and 83% accuracy for road images.