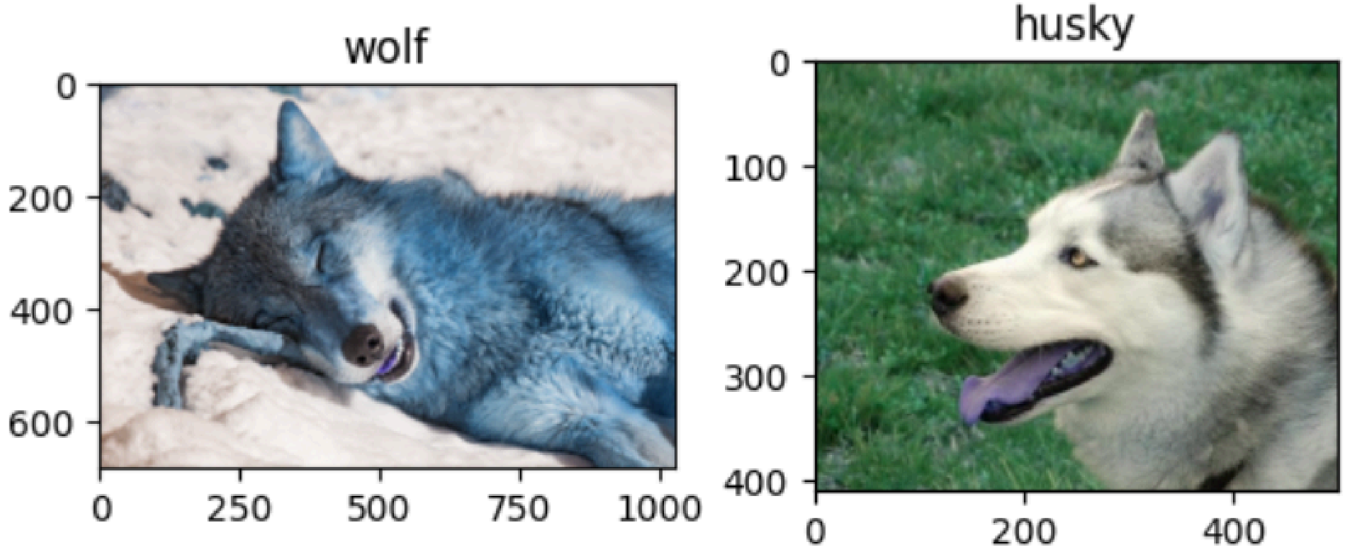
Alexandria University
Faculty of Engineering
Computer and Communications program

Spring 2025
CSE357 Pattern Recognition
Assigned: 28th of February

Assignment 1

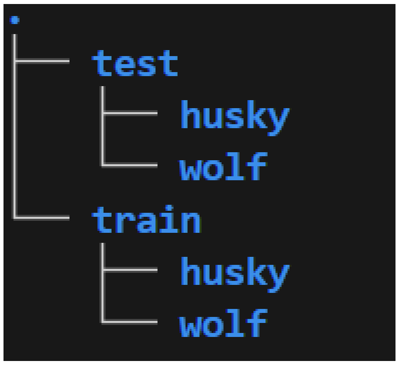
Husky Vs Wolf:

This is a binary classification task involving two classes: husky and wolf. Your goal is to develop a classification pipeline that can process images of any size and determine whether each image depicts a husky or a wolf.



Data Loading (10 points)

The dataset is organized as split/class/images. You need to load the dataset and labels accurately. To verify this, plot some image-label pairs.



```
.
├── test
│   ├── husky
│   └── wolf
└── train
    ├── husky
    └── wolf
```

Data Splitting (10 points)

Split the data into training and validation sets. You can choose the ratio you find most suitable. This split can also be done as part of cross-validation if you decide to use k-fold cross-validation to get a better model.

Data Preprocessing (30 points)

The raw data cannot be used directly; it requires preprocessing. This should include:

- **Resizing:** Ensure all images are of the same size.
- **Normalization:** Scale the pixel values to a range suitable for the model.
- **Augmentation:** Apply transformations such as rotation, flipping, and cropping to increase the diversity of the training set.

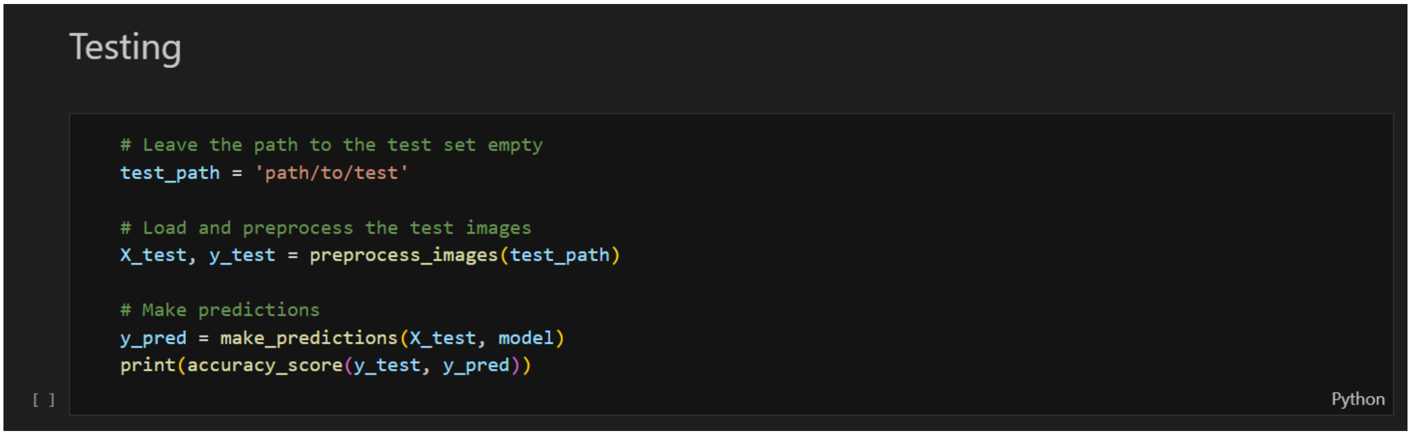
You can also do any other thing you would like to ensure the data is in an optimal state for the model to learn effectively but you **shouldn't use pre-trained models** in this step. Remember, pre-processing is a crucial step and can significantly impact the model's performance and accuracy.

Creating the classifier (40 points)

For this section, you must create either a Logistic regression or a neural network model. Hyperparameter tuning is required, and you should include the results in the report in the form of a table of different combinations and the results you were able to get. You may include additional models, as long as they do not affect the overall runtime of the notebook during grading. If you add models solely for discussion purposes, please comment out the relevant code and include the results in your report.

Correct Evaluation (10 points)

Your final submission must include a placeholder for testing, allowing the addition of the test data path. Upon using the "Run All" option in the notebook, the grading process will rerun the entire training process and evaluate the results on the test data. If you want to submit a python file instead of a notebook, make sure to add an argument for the test path.



Testing

```python
# Leave the path to the test set empty
test_path = 'path/to/test'

# Load and preprocess the test images
X_test, y_test = preprocess_images(test_path)

# Make predictions
y_pred = make_predictions(X_test, model)
print(accuracy_score(y_test, y_pred))
```

Bonus 1: Using Grad-CAM (10 points)

Grad-CAM is a technique used to generate a heatmap that highlights the areas of an image the model focuses on during prediction. It's important to understand how to utilize Grad-CAM and to plot at least four samples, demonstrating and explaining the specific parts of the image that influence your model's predictions. **You might want to try using a CNN for this part.**

Bonus 2: Detect and Fix the data leakage (10 points)

The provided dataset contains a clear case of data leakage that you can address. To earn the bonus, you should explain the data leakage, demonstrate how to fix it and Create a dataset of at least 10 images to test the model, taking the data leakage into account.

Notes:

- You are allowed to work in teams of up to 3.
- The dataset size is small to allow numerous trials since this is your first assignment.
- The test set will not be provided.
- Submit a notebook/file with your code and a report detailing your work.
- Teams may lose credit in the discussion if any member has not contributed.
- Any cases of cheating will result in a score of 0% for all involved teams.