

# CSE4/510: Applied Deep Learning Summer 2020

**Instructor: Alina Vereshchaka**

## Project 3 -- Image Classification

**Deadline: July 28, Tue, 11:59pm**

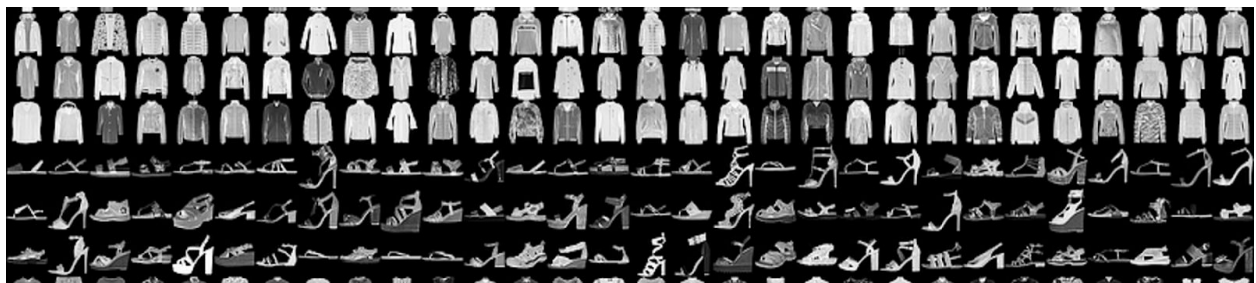
### Description

In this assignment we will practice to train Convolutional Neural Networks on Fashion-MNIST and ImageNet datasets. The goals of this assignment is to explore CNN models, techniques for preventing overfitting and various data augmentation tools. We will also get practice on working with one of the key datasets in computer vision.

### Dataset

For this project we will be working with two datasets:

#### Fashion-MNIST



[Fashion-MNIST](#) consists of a training set of 60,000 examples and a test set of 10,000 examples. Each example is a 28x28 grayscale image, associated with a label from 10 classes.

#### Getting the data:

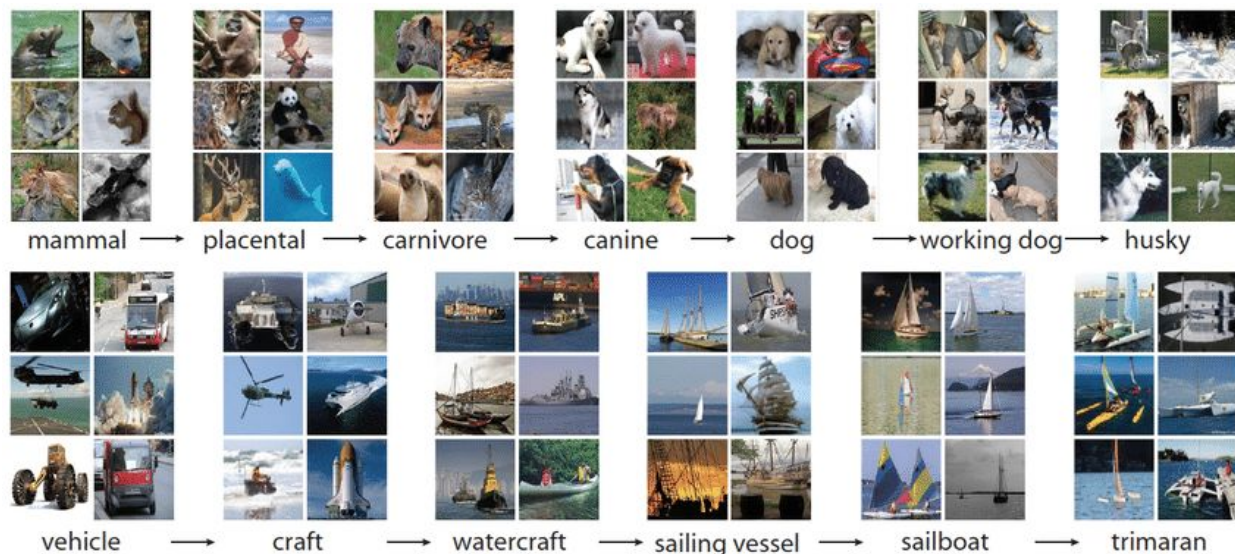
- Keras API: [https://keras.io/api/datasets/fashion\\_mnist/](https://keras.io/api/datasets/fashion_mnist/)

Example:

```
from keras.datasets import fashion_mnist
(trainX, trainy), (testX, testy) = fashion_mnist.load_data()
```

- Pytorch: <https://pytorch.org/docs/stable/torchvision/datasets.html#fashion-mnist>
- Direct download: <https://github.com/zalandoresearch/fashion-mnist>

## ImageNet



[ImageNet](#) is a large visual database with more than 14 million human annotated images and 20k classes designed for developing computer vision algorithms. ImageNet Large Scale Visual Recognition Challenge (ILSVRC) led to developing a number of state-of-the-art algorithms. For the project we would work on a limited number of classes.

### Getting the data:

- List of all classes can be found [here](#)
- Linux/MacOS: <https://github.com/mf1024/ImageNet-Datasets-Downloader>
- Windows: <https://github.com/skaldek/ImageNet-Datasets-Downloader>
- [Direct download](#) (~150GB). For the Project 3 full dataset is not needed, we will be working with a limited number of classes.

## Tasks

### Part I: Fashion-MNIST Classification [20 points]

1. Upload Fashion-MNIST dataset and prepare for training (normalize, split between train/test/validation).
2. Build a ConvNet with at least 3 convolutional layers.
3. Discuss the results and provide the graphs, e.g. train vs validation accuracy and loss over time. Show the confusion matrix.

### Part II: Data augmentation and CNN improvements [40 points]

1. Increase the dataset by x4 using any data augmentation techniques (rotations, shifting, mirroring, etc). You can use a combination of these techniques simultaneously.
2. Apply tools that help to prevent overfitting (regularisers, dropouts, early stopping, etc). Discuss each of them and how they impact the testing performance.
3. Discuss the results and provide the graphs, e.g. train vs validation accuracy and loss over time. Show the confusion matrix.

### Part III: ImageNet Classification [40 points]

1. Choose at least 20 classes from ImageNet. Each class has to contain at least 500 images.
2. Download the dataset (see above for the links to downloader scripts).
3. Preprocess the dataset for training (e.g. removing missing images, normalizing, split between training/testing/validation)
4. Build a CNN classifier with at least 3 convolutional layers to train on an ImageNet dataset that you have collected. You may use the same model as for Part 2.
5. Discuss the results and provide the graphs, e.g. train vs validation accuracy and loss over time. Show the confusion matrix.

## Submit the Project

- Submit at **UBLearn**s > **Assignments**
- The code of your implementations should be written in Python. You can submit multiple files, but they all need to have a clear name

- Upload your ImageNet created dataset to UBox or Google Drive and include the link in your report
- All project files should be packed in a ZIP file named **TEAMMATE#1\_UBIT\_TEAMMATE#2\_UBIT\_project3.zip** (e.g. **avereshc\_neelamra\_project3.zip**).
- Your Jupyter notebook should be saved with the results. If you are submitting python scripts, after extracting the ZIP file and executing command `python main.py` in the first level directory, all the generated results and plots you used in your report should appear printed out in a clear manner.
- In your report include the answers to questions for each part. You can complete the report in a separate pdf file or in Jupyter notebook along with your code.
- Include all the references that have been used to complete the project.

## Important Information

This project can be done in a team of up to two people. The standing policy of the Department is that all students involved in an academic integrity violation (e.g. plagiarism in any way, shape, or form) will receive an F grade for the course. Refer to the [Academic Integrity website](#) for more information.

## Late Days Policy

You can use up to 5 late days throughout the course that can be applied to any project. You don't have to inform the instructor, the late submission will be tracked in UBlerns. If you work in teams the late days used will be subtracted from both partners. E.g. you have 4 late days and your partner has 3 days left. If you submit one day after the due date, you will have 3 days and your partner will have 2 days left.

## Important Dates

**July 28**, 11:59pm - Project 3 is Due