ASSIGNMENT II
Convolutional Neural Networks

SH2150 Machine Learning in Physics

November 2025

In this assignment, you will investigate the progression from hand-crafted image features to deep learned representations using convolutional neural networks (CNNs). You will work with one of the provided datasets (`galaxies`, `brain-stroke`, or `histopathology`), each containing three classes. Each dataset `.zip` file contains more information about the respective data.

Your tasks are to

- extract hand-crafted features from images,
- perform unsupervised clustering and analyze the results,
- train a feed-forward neural network on the extracted features,
- train a convolutional neural network classifier using the full images,
- compare the performance of the models using confusion matrices and t-SNE plots.

Hint: Normalization of Features Before training your models, it is important to **normalize** the features. Different features may have very different scales (for example, mean pixel intensity may range from 0–255, while variance or edge density may lie in a much smaller range). Models such as K-means and MLPs are *sensitive to feature scaling*, and unnormalized data can lead to poor clustering and slow or unstable training. Normalization ensures that all features contribute comparably during learning. A common approach is to apply **standardization** (subtract the mean and divide by the standard deviation for each feature), computed across the training set and then applied to both training, validation and test data. Another approach is **min-max** normalization, where you map all features to the range $[0, 1]$ or $[-1, 1]$.

1. Hand-crafted Feature Extraction and Unsupervised Analysis
   (a) Extract 3-5 meaningful but simple hand-crafted features (e.g. intensity mean, standard deviation, edge density, etc.) from each image in your dataset. Briefly describe your choices and expected usefulness for classifying the images into the three categories. (2p)
   (b) Perform K-means clustering (either define it yourself or use the `KMeans` algorithm from `sklearn.cluster`) with $K = 3$ clusters on the training data. Use majority-voting on the labels of the fitted data points in each cluster to assign a label to it. (2p)

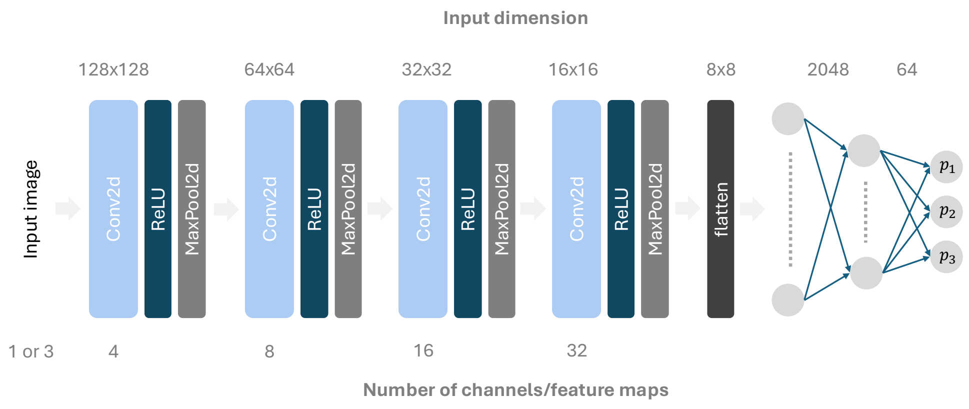

Figure 1: CNN architecture example. The network consists of four convolutional blocks consisting of a Conv2d layer with *same* padding, ReLU activation, followed by a MaxPool2d layer respectively. The final (MLP) part of the network maps the flattened output from the CNN blocks through a first fully connected (linear) layer followed by a ReLU activation. The final fully connected layer is followed by a softmax activation to produce the final class probabilities p_1, p_2, p_3.

(c) For each point in the test set, predict what cluster/category it belongs to and visualize the data using t-SNE in $\mathbb{R}^2$. Discuss how well clusters align with true labels. Also, report the confusion matrix. (2p)

2. MLP Classifier on Extracted Features

   (a) Define an MLP classifier similar to the network in the previous assignment sheet (Assignment I). Make sure the output dimension is three and switch to `torch.nn.Softmax()` activation in the final layer. Use one-hot encoded targets. You might want to play around a bit with the hidden activation functions (not the final one). (2p)

   (b) Train the MLP using cross-entropy and the Adam optimizer, on the extracted features in 1(a) from the train set. Validate the model performance on the validation set during training; plot a loss curve for the train set and val set respectively, as a function of epoch. Does the model generalize well? (2p)

   (c) Evaluate the model on the test set and report a confusion matrix. For each data point, extract the output from the final layer of the network, project it to $\mathbb{R}^2$ using t-SNE and report a plot. Which classes were hardest to distinguish and why? (2p)

3. CNN Classifier on Full Images

   (a) Compare the number of trainable parameters in the following two layers. Assume the image is single–channel (grayscale). (1p)

   - A fully connected (linear) layer that takes the flattened 128×128 image as input and outputs 512 neurons.
   - A convolutional layer with 64 filters, each of spatial size 3×3 (stride 1, padding such that the spatial dimensions are preserved).

   For each case, write the formula you use and compute the total number of parameters. Finally, explain what your result implies about the risk of overfitting when using fully-connected layers versus convolutional layers on images.

(b) Implement and train a CNN classifier. The CNN should input the full images (128×128 pixels) and output one probability for each class, similar to the MLP in exercise 2 above. The architecture can be defined according to figure 1, but you are free (not mandatory) to design your own architecture if you describe your design choices. Each convolutional layer should be followed by a non-linear activation function like `ReLU`, and a 2×2 max-pooling. In the final stage of the network, the features are flattened and two fully connected (linear) layers are used to produce the class probabilities. For training, use cross-entropy loss, Adam optimizer, and validate the model performance on the validation set in each epoch. Provide loss curves as a function of epoch, as in 2(b). (2p)

(c) Evaluate the performance of your CNN classifier by first computing the confusion matrix on the test set and briefly commenting on which classes are most often misclassified. (1p)

(d) For each convolutional block in the network, extract the feature representations of all samples in the validation set and use t-SNE to project these features into $\mathbb{R}^2$. Create one two-dimensional visualization per block. Describe how the structure of these feature spaces evolves through the network, with particular attention to how the intra-class compactness and inter-class separability change as depth increases. (2p)

(e) Finally, compare these observations to the separability observed in the hand-crafted feature space used for K-means, as well as to the learned feature space of the MLP classifier. Discuss why the CNN may achieve stronger class separation and more robust classification performance. (2p)

Final question: Did you use an AI tool (other than the machine learning models you trained in this exercise) for anything else than information searching, when solving this problem set? If so, please write a **brief statement of how you used AI.**

Total number of points: 20

Remember to motivate your answers wherever applicable.

Good luck!