Introduction to Deep Learning – Final Project 2024/2025

🎨 Bring Van Gogh Back to Life! 🎨

Submissions due: February 2nd, 2025.

Instructions for accessing the data and working in Lab 424 are provided on the last page.

Overview:
Over the past few weeks, you've delved into the exciting world of deep learning, uncovering the magic behind neural networks. Now, it's time to put your knowledge into action with a creative twist!

Imagine this: You've just learned about **style transfer**, and suddenly, a flash of inspiration strikes. You recall the legendary artist **Vincent van Gogh**, whose mesmerizing paintings like *Starry Night* and *Sunflowers* are celebrated across the world. Now, imagine harnessing the power of neural networks to recreate Van Gogh's iconic style. also, your chance to not only showcase your technical skills but also your artistic flair!

Focus on these two pre-trained models: **VGG-19** and **AlexNet**.

Importing the pre-trained models (as was shown in exercise 6):

```
from torchvision import models

vgg19 = models.vgg19(pretrained=True)

alexnet = models.alexnet(pretrained=True)
```

Part 1

Your first step is to fine-tune the pre-trained models (VGG19 and AlexNet) on Post-Impressionism data. This dataset is a subset of the larger WIKIART collection.

Goal:
Fine-tune both models to perform binary classification: determining whether a painting is painted by Van Gogh (is_van_gogh: yes/no).

Preparing the Data (read the last page of this document!!):

- In the provided link/ the directory in lab 424, you'll find a CSV file – "classes". This file contains metadata for each image in the dataset, including the artist's name and the filename of the corresponding picture which should be used to extract labels for your data. Each row represents one image.
- Don't forget to apply the proper transformations.

Augmentation:

- Apply a variety of data augmentation techniques during training to create a more robust model. For example, consider augmentations like rotations, flips, color adjustments, and cropping.

HP-tuning:

- Use Optuna. Hint: watch exercise 7.

Tracking Experiments:

- Use tools like Weight & Biases to track your experiments.
- Log details like the augmentation techniques used, hyperparameters, and model performance for better reproducibility and analysis. Include them all in the final report. Add explanations to the experiment's results you are getting.

Cross-Validation:

- Implement K-Fold Cross-Validation to improve the generalization of your model and avoid overfitting.

Performance Metrics:

- Evaluate the model using metrics such as **Accuracy** (Overall performance), **AUC-ROC** (Ability to distinguish between classes), **F1 Score** (Balance between precision and recall) and so on…

Creative Additions:

- Feel free to add your ideas or innovative solutions to improve the model. Your creativity is encouraged and will affect your grade!

Part 2 - In this section, you will create a style transfer function.

Required Inputs:

- **Model**: The pre-trained neural network to use for style transfer.
- **Predefined style layers**: Layers of the model responsible for extracting style features.
- **Predefined content layer**: The layer responsible for capturing content features.
- **Weights for style layers**: Individual weights to balance the contribution of each style layer.
- **Weight for overall style intensity**: A scalar value to control the magnitude of the style.
- **Weight for content intensity**: A scalar value to control the emphasis on content.
- **Content image**: The image whose structure you want to preserve.
- **Style image**: The image whose artistic style you want to apply.

Output:

- The function should return the final processed image with the desired style applied.

The function should be generic and compatible with both models.

Steps to Follow:

1. **Apply the Function**:
    - For each model, use your style transfer function on 20 images of your choice. Personal photos are a great option for a more enjoyable experience.
2. **Set Epochs**:
    - Decide on an appropriate number of epochs. This number must remain consistent across all images and models.
3. **Select Style Images**:
    - Use at least 5 Van Gogh paintings as style images.
    - OPTIONAL: you can combine a few styles from different Van Gogh paintings.
4. **Evaluate Quality**:
    - Use both of your previously built classification models to assess the quality of the style transfer – include a detailed discussion in the final report.
5. **Compare Models**:
    - Evaluate and compare the performance of the style transfer between the models, using metrics of your choice.

Discussion

Analyze your results and discuss them with respect for the following questions.

** Note that some requirements were detailed in the sections above, make sure you don't forget to include anything.

Part 1:

- **Describe the Dataset**: Provide an overview and analysis of the dataset and determine what proportion of it consists of Van Gogh's paintings.
- **Detail the Models**: Explain the models you are using, highlighting their unique characteristics, the datasets they were originally trained on, their architectures, publication dates, and any other relevant distinctions. Describe the transformations you applied on the provided data set accordingly.
- **Explain the Training Method**: Outline the approach you used for training, including the types of augmentations applied, additional layers added, use of transfer learning, optimizer choices, and any other relevant details/ decisions you made during the project.
- **Compare the Models**: Analyze and compare the two models, discussing the nature of their predictions, their strengths and weaknesses, and overall effectiveness.
- **Evaluate Predictions**: Present examples of true positives, true negatives, false positives, and false negatives for each model. Provide the full confusion matrix. Provide an analysis of these examples to understand where the models perform well and where they may need improvement.

Part 2:

- **Describe the Style Transfer Process**: Explain the mechanics of style transfer, detailing how content and style features are extracted and combined to produce the final image. Connect your explanations to the function you have built and attach a screenshot of it to the report.
- **Importance of Normalizing Style Loss Layers**: Discuss why it's necessary to normalize each style loss layer and its impact on achieving balanced style transfer.
- **Analyze Examples** (using the same content image):
  - Cases where **both models** recognize the result as a Van Gogh painting.
  - Cases where **one model** recognizes it as Van Gogh, but the other does not (and vice versa).
  - Cases where **neither model** recognizes it as Van Gogh.

- **Layer Selection**:
  - Specify which layers from each model were chosen for content and style extraction.
  - Justify your choices based on the purpose of each layer and its role in style transfer.
- **Classification Model Choice**:
  - Identify which classification model you used to evaluate the nature of the style transfer differences between the two models.
  - Explain why this model was selected.
- **Model Validation**:
  - Analyze whether the classification model confirms the differences observable to the naked eye.
- **Results Comparison**:
  - Reflect on how the results from this part align or differ from the findings in part 1 and discuss their implications.

Your explanations and discussions will significantly impact your grade.

Since the base code for style transfer (from Dor's presentation) and fine-tuning a pretrained model (Exercises 6 and 7) were provided, we are particularly interested in your ability to demonstrate understanding of the code, analyze the graphs in W&B, grasp the project flow, and articulate the underlying concepts and ideas and so on... Show us that you understand.

Submission Instructions:

1. **Report**:
   - Write the report in a **Word document and submit it as a PDF**.
   - Include all relevant results, images and graphs at the end of the document as **appendices**.
   - Specify which **non-standard Python packages** were used, explain their purpose, and why they were chosen.
   - Separate all the required discussion parts in a way that it will be easy to connect your answers to the different points.
2. **Code Submission**:
   - Ensure all processes implemented in the code are traceable for evaluation.
   - Submit files ONLY in .ipynb format!
   - Attach the **Jupyter Notebook** (or notebooks if you have split the process) with all **outputs saved** and **visible** (ensure that all cells are executed, and outputs are displayed).
   - * Note - a link to a GitHub repository (that contains the jupyter notebooks) is also acceptable.
3. **File Format**:
   - Submit both the PDF file and Jupyter Notebook in a single compressed file (.zip or .rar).
   - Name this compressed file Group_XX_DL_Project (XX= group_num)
4. **Optional Enhancements:**
   - If additional files, resources, or tools were used, include them as supplementary material in the same compressed file.

lab 424 instruction

1. open anaconda navigator.
    a. Go to environments.
    b. Click import and then choose a local device.
    c. The environment is under:

       "R:\DeepLearning_env.yaml"

    d. After loading the environment, right click on the green arrow, and choose ‘’open with terminal’’
    e. Copy this command to terminal and wait for the process to finish:

       ```
       pip3 install torch torchvision torchaudio --index-url
       https://download.pytorch.org/whl/cu124
       ```

2. Accessing the data:
    a. On the R drive you have a directory called DeepLearning2024, inside you have the csv of the metadata and the directory of the photos.
    b. The larger dataset is available on Kaggle website, but further filtering will be required in order to work with it directly from Kaggle (as described below).
3. Fine-tuning the model:
    a. To efficiently finetune the model, it is recommended to do so using PyCharm (it will be 3-4 times faster than in a notebook).
    b. You can save the trained model and upload it to the notebook, in which you can display your outputs and continue coding easily.
    c. Notice- attach the relevant W&B reports and printed outputs to your report.

Home instructions:

If you have access to a GPU at home or through a remote account (and don’t want to come to lab 424), you are welcome to use it for your project.
There are two options for accessing the data:

1. Go once to lab 424, download the provided files once, keep them in google drive (or any other platform) and then use it on your machine.
2. GO to the provided link: WIKIART and filter the data to contain **Post impressionism** data only. Then, use the csv to link the pictures and the labels.

After having the properly filtered data, continue working on your GPU as you wish :)