DD2424, 2025 - Defining your project

Teachers and TAs of DD2424 2025

It might be daunting, given the amount of material on the web, to try and narrow down the scope and subject matter of your project. In this document we will help you achieve this by suggesting some interesting projects that you could complete and where labelled datasets exist. In particular we define two categories of projects you can choose from.

One is the set of **default projects** where each group builds the same model to ensure a strong baseline and the group will receive an E grade if they achieve a sufficient level of performance and write a clear and decent report. Within the default project a group can then aim for a higher grade if you complete an extension and/or extensions from the baseline. The increase in grade depends on the level of ambition of the extension(s), the quality of the implementation and the report. The other project type is the **custom project** where you define your own project and the grade once again depends on the level-of-ambition of the project and the quality of the experimentation and the report.

Ideally we would like students who do not have so much software or deep learning experience to complete a default project, but if your group is relatively experienced with deep learning and/or software engineering then we see no problem with you defining a custom project.

Regardless of which project type you complete, your group will still need to complete a project proposal form.

1 Default Projects

The titles of the three default projects are:

1. Explore Transfer Learning
2. Building and training a modern ConvNet architecture from scratch
3. Text Generation

If you visit the Canvas Project page you can download a detailed description of each project.

2 Custom Final Project

2.1 General guidelines for choosing a project topic

What recognition, synthesis, manipulation or translation (or some other) task interests you? Is there a network architecture or training algorithm that you would love to investigate in further

detail? Is there a way you can explore this interest in a meaningful yet computational feasible way? Or is there a particular application that interests you and can potentially be tackled/solved by deep learning? If you think the answer is yes to any of these questions, then you may have found the topic for your project. In all cases you should study the literature and the plethora of tutorials and blog posts on the web regarding your project idea. Find out the most common and successful deep learning solution to your project topic. If it's an architecture or training algorithm you're investigating then find out what types of problems this network is used to solve and what datasets exist that you can use for your experiments.

You can replicate the results of a published paper, however, you need to do so in a *questioning* manner (what component of the introduced method was crucial to the final performance, can the method be applied to different data than in the original paper, ...). For any project you complete you need to define some relevant questions on the different aspects of the deep learning approach you chose. Conduct meaningful experiments regarding those questions and make and discuss the conclusions that can be made from these experiments. The set of questions can include trying the various techniques taught during the lectures. But you are not limited by the content of the lectures. **We will not accept, though, projects with a focus on re-inforcement learning. This topic is out-of-scope for this course so we will not allow them.**

2.2 Your own specification

You are absolutely free to choose any task or idea you are excited about and a corresponding dataset. But we would advise that you use some paper (from a reputable source) to help guide your project. You should also consider the size of your dataset relative to your computational resources so it is fine to downsample the dataset to make working on it feasible. For links and ideas for different datasets please refer to the extra resources section below.

2.3 Paper Ideas: Aiming for grade $\geq B$

We assume project groups completing a custom project are aiming for a grade greater than $\geq B$. If this is not the case please review the default projects and the suggested extensions to get an idea of the level we expect for grades E-C.

The following are some suggestions (a drop in the ocean) of papers that could form the basis of your project and we would anticipate that the following papers would be good basis for students aiming to get a grade $\geq B$ and would be feasible to re-implement in the time-frame of the project. Many of the papers below have implementations available online. However, we expect each group to write most of the code themselves. Yes you can refer to online implementations to help clarify implementation details etc. But, one of the goals of the project is to get comfortable using modern deep learning packages so that you can complete your own projects after the course.

But, of course, we are also very aware that you may not be able to replicate all the experiments on all the datasets (especially the big ones) given the time and computational resources available to you! So get advice from the TAs about what is realistic.

- **Better understanding of convolutional networks**

- The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks by J. Frankle and M. Carbin published at ICLR, 2019

- **Batch Normalization and other normalization scheme**

  - How Does Batch Normalization Help Optimization? by S. Santurkar, D. Tsipras, A. Ilyasa and Aleksander Madry, NeurIPS, 2018.
  - EvalNorm: Estimating Batch Normalization Statistics for Evaluation by S. Singh and A. Shrivastava published at ICCV, 2019
  - Norm matters: efficient and accurate normalization schemes in deep networks by E. Hoffer, R. Banner, I. Golan and D. Soudry, NeurIPS, 2018

- **Semi-supervised learning & self-supervised learning**

  - MixMatch: A Holistic Approach to Semi-Supervised Learning, D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver and C. A. Raffel, NeurIPS 2019
    FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence, NeurIPS 2020
  - Unsupervised Representation Learning by Predicting Image Rotations by Spyros Gidaris, Praveer Singh, and Nikos Komodakis, ICLR 2018.
  - Self-Supervised Learning of Pretext-Invariant Representations by I. Misra and L. van der Maaten, CVPR 2020
  - Exploring Simple Siamese Representation Learning, CVPR 2021
  - A Simple Framework for Contrastive Learning of Visual Representations, ICML, 2020
  - Masked Autoencoders Are Scalable Vision Learners, CVPR 2022 (The Transformer version)

- **Some crazy data-augmentation technique**

  - MixUp: BEYOND EMPIRICAL RISK MINIMIZATION by H. Zhang, M. Cisse, Y. N. Dauphin and D. Lopez-Paz, ICLR 2018
  - CutMix: Regularization Strategy to Train Strong Classifiers With Localizable Features, ICCV, 2019

- **Learning with noisy labels**

  - Symmetric Cross Entropy for Robust Learning With Noisy Labels by Y. Wang, X. Ma, Z. Chen, Y. Luo, J. Yi and J. Bailey, ICCV 2019

- **Memory and computationally efficient networks**

  - ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices by X. Zhang, X. Zhou, M. Lin and J. Sun, CVPR, 2018
  - Distilling the Knowledge in a Neural Network by Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean, NIPS Deep Learning and Representation Learning Workshop (2015) (It might be fun to distill a largish pre-trained model to a much smaller ConvNet and see if works better than training from scratch.)

- TRAINING WITH QUANTIZATION NOISE FOR EXTREME MODEL COMPRESSION by Angela Fan, Pierre Stock, Benjamin Graham, Edouard Grave, Rémi Gribonval, Hervé Jégou, Armand Joulin, ICLR, 2021. (At the moment model quantization, especially for LLMs, is a hot research topic! Check out this page, Awesome-Quantization-Papers to see the recent activity in the area.)

- **Continuous Image representations**

  - Implicit Neural Representations with Periodic Activation Functions by V. Sitzmann, J. N. Martel, A. W. Bergman, D. B. Lindell, and G. Wetzstein, In Proceddings of NeurIPS, 2020
  - Learning continuous image representation with local implicit image function by Y. Chen, S. Liu, and X. Wang, In Proceddings of CVPR, 2021

- **Leverage Large Language / Vision Models to compensate for lack of training data**

  If you have a particular problem without much data, you can potentially exploit large pre-trained models, both multi-modal and uni-modal, to expand this data. A very straight forward example for an image classification system would be use a powerful pre-trained vision generation and language models to generate new images of a class. You could then compare this approach to data-augmentation to more traditional forms based on geometric transformations. Or perhaps to generate images for the case you are tackling the zero-shot problem. Note if you go in this direction please, please check if it is computationally feasible for your memory and compute resources.

- **Semantic Segmentation**

  - U-Net: Convolutional Networks for Biomedical Image Segmentation by O. Ronneberger, P. Fischer, and T. Brox
  - Label-Efficient Semantic Segmentation with Diffusion Models by D. Baranchuk, I. Rubachev, A. Voynov, V. Khrulkov, and A. Babenko, ICLR 2022. The paper explores how to extract an image feature representation from a U-Net pre-trained as a diffusion image generator and then investigates using this for the task of semantic segmentation. Here you replicate some of the ideas explored in the paper and perhaps also tweak them with your own idea and you can, of course, reply on an appropriate pre-trained U-Net for your project.

- **Auto-encoders**

  - Memorization in Overparameterized Autoencoders by A. Radhakrishnan, M. Belkin, C. Uhler, ICML, 2019

- **Improving performance of Convolutional network**

  - Attention Augmented Convolutional Networks by I. Bello, B. Zoph, A. Vaswani, J. Shlens, Q. V. Le, ICCV, 2019
  - Cbam: Convolutional block attention module. by S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, ECCV, 2018

- Drop an Octave: Reducing Spatial Redundancy in Convolutional Neural Networks With Octave Convolution by Y. Chen, H. Fan, B. Xu, Z. Yan, Y. Kalantidis, M. Rohrbach, S. Yan, J. Feng published at ICCV, 2019
- Information Entropy Based Feature Pooling for Convolutional Neural Networks by W. Wan, J. Chen, T. Li, Y. Huang, J. Tian, C. Yu and Y. Xue published at ICCV, 2019

- **Improving network learning by adjusting training details**

  - Soft Augmentation for Image Classification by Yang Liu, Shen Yan, Laura Leal-Taixeé, James Hays, Deva Ramanan, CVPR, 2023 . This paper describes a simple but effective idea. When you apply strong augmentation then make the targets in the cross-entropy loss more ambiguous. Try out the idea on Cifar10 and/or Cifar100 and see how it works for you!
  - Long-tailed recognition via weight balancing Shaden Alshammari, Yu-Xiong Wang, Deva Ramanan, Shu Kong, CVPR, 2022. Supervised learning with cross-entropy loss and imbalanced data does not in general produce great results for the minority classes. The most recent trend when you have imbalanced data is to have a two stage process. The first stage is to do normal training to learn the image encoder and then to train the classifier based on this encoder. In the second step class-imbalance losses or strategies are applied during training. This paper is interesting as it highlights that it is really important to not let the magnitude of your network's grow large if you want good performance. Prioritise the strategies in the paper that give you the most bang for your buck and use the cifar100-lt dataset and a small ResNet for experimentation.

- **Image Generation with Diffusion Models**

  - Implement and train an image diffusion model to generate images. Use a relatively simple dataset such as Cifar10 or the Oxford Flower dataset. The most straightforward approach and computationally efficient would be to follow the tutorial outlined in Denoising Diffusion Implicit Models based on the Denoising Diffusion Implicit Models by Jiaming Song, Chenlin Meng, Stefano Ermon, ICLR 2021. This approach is more computational efficient for generation than the original Denoising Diffusion Probabilistic Models by Ho et al., NeurIPS 2020 but the training process is very similar. The main difference is in the noise scheduling.

- **Transformer Networks**

  - Train a VIT like network on Cifar-10/Cifar-100 and see how it goes! At the moment ViT type models trained from scratch on small images and relatively small datasets have worse performance than ConvNets trained from scratch on the same datasets. But this exercise would be more about getting into the details of ViT architecture.
  - In 2021 there was a flurry of work investigating if the attention layers in ViT are strictly necessary. In particular they replace the attention layer with some other operation (without the quadratic complexity) to allow communication between the representations of different patches. Here are two such papers:
    - MLP-Mixer: An all-MLP Architecture for Vision by I. Tolstikhin et al., NeurIPs 2021.

- Do You Even Need Attention? A Stack of Feed-Forward Layers Does Surprisingly Well on ImageNet by Luke Melas-Kyriazi, arxiv 2021

One can interpret the architectures described in these papers as very particular forms of ConvNet.

- In a similar vein to the prior papers there is the extension to this work is the ConvMixer described in Patches Are All You Need? by A. Trockman and J. Zico Kolter. The architecture can be trained on Cifar-10 from scratch more easily to have good performance than the MLP-Mixer. It might also be fun to investigate using a pre-trained version of this architecture to perform semantic segmentation as no downsampling, apart from the patch extraction at the start, is performed in the network.

3 Resources

3.1 Publicly available computer vision datasets

Here are some classic visual recognition and classification tasks with a corresponding dataset:

- Scene classification: To classify a given image into different scene labels. Dataset: Places
- Semantic segmentation: To classify every pixel of an image into a class. Dataset: MS COCO
- Human detection: To detect the location of all the humans in an image (if any). Datasets: Caltech, Daimler
- Pose estimation: To estimate the location of the joints of a given person. Dataset: MPII
- Video classification: Classify a video into a set of pre-defined classes. Dataset: Youtube Sports
- Depth estimation: Estimate the depth at each pixel from the RGB data. Dataset: NYU
- Gaze estimation: Estimate the eyes gaze of a person based on his image. Dataset: MPII Gaze

The webpages CV Datasets on the web and Yet Another Computer Vision Index To Datasets (YACVID) have a more comprehensive list of available datatsets.

3.2 Resources for Natural language processing

NLP is a domain where variations of RNNs and Transformer networks come into their own. The site Hugging Face has many pre-trained models available for downloading. You could potentially use some variation of RNN/transformer network to perform some form of text classification, generation and/or translation. The excellent site spro/practical-pytorch has several links to several tutorials, such as Practical PyTorch: Translation with a Sequence to Sequence Network and Attention or Practical PyTorch: Classifying Names with a Character-Level RNN exercise suggestions. These tutorials and exercise suggestions could definitely form the basis for interesting projects. The Stanford course CS224n: Natural Language Processing with Deep Learning is potentially also another great resource and source of project ideas of a more sophisticated and ambitious nature.

3.3 Resources for Speech/Sound

Unfortunately, my hands-on experience in speech recognition and processing is very limited. But I'm very open to finding out more by reading your projects in the area. Here are two Speech Recognition datasets that may be of practical help:

- CMU Robust Speech Recognition Group: Census Database

Or here is a link to code, a dataset and paper description of how to use a LSTM to *compose* folk music Folk music style modelling using LSTMs. It is unclear to me how long training would take in this case.

4 Popular software packages for deep learning

These frameworks are currently the most popular among academic researchers. (Most take advantage of Nvidia's deep learing libraries so their computational timings do not differ that much.)

- PyTorch: Maintained collaboratively by people at Facebook, NYU, ParisTech, Nvidia, ... Written in `python`.
- TensorFlow: The open-source Google deep learning framework. It has both a `C++` and `python` API.
- JAX: This is NumPy with automatic differentiation which you can run on CPU, GPU and TPU. It also has just-in-time compilation. It is maintained by Google. Flax is the neural network library for JAX and it is an open source project maintained by Google Research.
- Caffe: created by Yangqing Jia and maintained by BVLC at Berkeley and open-source community. In `C++` and `Cuda` with limited python and MATLAB wrappers.

If I've missed a popular package, please let me know and I can add it. You are free to choose whichever package you like and feel most comfortable with.

5 Top-tier conferences for inspiration

Here are some top-tier conferences where you can find relevant papers from the fields of computer vision and deep learning:

- Main focus **Computer vision**
  i) ICCV ii) ECCV iii) CVPR iv) BMVC
- Main focus: **Spoken Language Processing**
  i) InterSpeech ii) EuroSpeech
- Main focus: **Natural Language Processing**
  i) InterSpeech ii) EMNLP

- Main focus: **Learning Representations**

  i) ICLR

- Main focus: **Machine Learning**

  i) NeurIPS ii) ICML

  Once you have identified a topic of interest, you can search Google Scholar with related keywords on an academic search engine: http://scholar.google.com