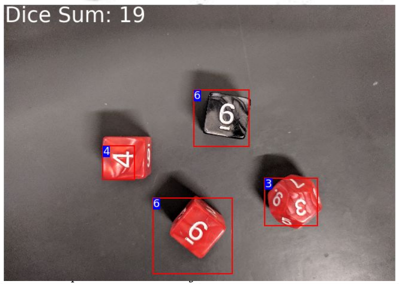
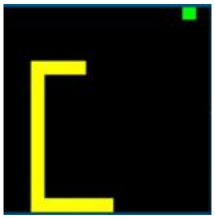
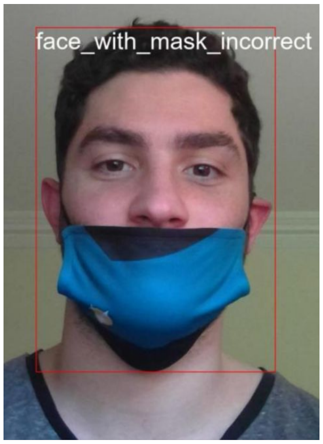
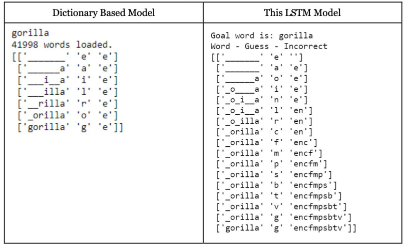
ECE4179/5179 - Deep Learning and Neural Networks

Course Project







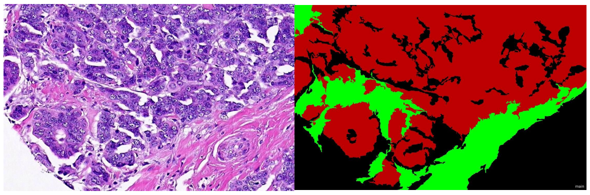


Dictionary Based Model

```
gorilla
41998 words loaded.
[['_______' 'e' 'e']
 ['______a' 'a' 'e']
 ['___i__a' 'i' 'e']
 ['___illa' 'l' 'e']
 ['__rilla' 'r' 'e']
 ['_orilla' 'o' 'e']
 ['gorilla' 'g' 'e']]
```

This LSTM Model

```
Goal word is: gorilla
Word - Guess - Incorrect
[['_______' 'e' '']
 ['_______' 'a' 'e']
 ['______a' 'o' 'e']
 ['_o____a' 'i' 'e']
 ['_o_i__a' 'n' 'e']
 ['_o_i__a' 'l' 'en']
 ['_o_illa' 'r' 'en']
 ['_orilla' 'c' 'en']
 ['_orilla' 'f' 'enc']
 ['_orilla' 'm' 'encf']
 ['_orilla' 'p' 'encfm']
 ['_orilla' 's' 'encfmp']
 ['_orilla' 'b' 'encfmps']
 ['_orilla' 't' 'encfmpsb']
 ['_orilla' 'v' 'encfmpsbt']
 ['_orilla' 'g' 'encfmpsbtv']
 ['gorilla' 'g' 'encfmpsbtv']]
```



1 Introduction

One of ECE4179/5179's main goals is to prepare you to apply deep learning algorithms to real-world tasks and prepare you to start AI research. The course project is intended to start you in these directions and demonstrates to us that you are now capable of conceptualizing, working out and implementing a DL solution to a problem beyond what is covered in the labs, and assignments. In other words, it is intended to make you exercise the skills you learn during the course on newer, and more exciting problems. We expect that a number of students continue working on their projects after completing ECE4179/5179, submitting their work to a conference or journal. We will encourage this and help students to achieve this goal.

Team. Each course project should be done by a team of max three students . You should only contribute to 1 course project. Even though you may complete the project individually, due to the work required we recommend you form groups. Training deep learning models can be very time consuming, so make sure you have the necessary compute in your team (we are negotiating with eResearch to support you but you should always be vigilant about this need).

Project Topics

Your first task is to pick a project topic. If you are looking for project ideas, please come forward and talk with us. We are happy to brainstorm and suggest some project ideas. Note that most students do one of the following type of projects:

- **Application project.** Pick an application that interests you, and explore how best to apply learning algorithms to solve it.
- **Theory project.** Pick a problem or family of problems, and develop a new learning algorithm, or a novel variant of an existing algorithm, to solve it.

We have seen projects combining elements of application and theory. A recipe for success is to pick either an **application** area that you are interested in, or to pick a subfield of deep learning that you want to explore more. So, pick something that you can get excited and passionate about! Be brave and bold, and do feel free to propose ambitious projects. If you are already working on a research or industry project that deep learning might apply to, then you may already have a great project idea. Alternatively, we will offer a default project if you are not able to define a project (details in 2). All projects must be suitably "novel" and contain a significant amount of genuine work. For example, "basic classification" tasks, anything too similar to labs or any simple implementation of an existing algorithm will not be resulting in a good mark. That said, note that we are **NOT** expecting ground-breaking research.

Assessment

Each project has three main elements, **1.** Project proposal, **2.** Project report, and **3.** Project presentation. The weight of each component is shown in table 3.

Task	**Weight**	**Notes**
Project Proposal	10%	See 1.1
Report	60%	See 1.2
Presentation	30%	See 1.3

Table 1: Course project - weight of each component

1.1 Project Proposal

By the end of week 9, you should submit a 1page proposal describing the following elements of your course project,

Title of your project	
Team members	
Methodology (include references)	You are supposed to narrow-down the approach you are going to take and have a baseline developed. In your proposal, you just need to clearly explain how you will approach the problem though. For example, “based on our literature review, we will use the method X developed by Y to tackle the problem Z. We have found an implementation of X at github-method-X, which we will use as the baseline for our work.”
Data	describe the dataset you will use to evaluate your work. Be extra careful with data for your project. If data needs considerable pre-processing to suit your task, or if you intend to collect the needed data yourself, keep in mind that we still expect a solid methodology and discussion of results.

Table 2: Course project - weight of each component

Grading. The project proposal is mainly intended to make sure you decide on a project topic and get feedback from us. As long as your proposal follows the instructions above and the project has been thought carefully, you should get its full mark.

1.2 Project Report

Your project report can be at most 5 pages long (including appendices, figures and references). For writing reports, see “Writing an Engineering lab report” or “Writing an Engineering technical report”. Your report should describe the following elements of your work;

1. **title of your project**
2. **team members.**
3. **Introduction.** Explain the problem and its importance. Discuss your motivation for pursuing this problem. Give some background if necessary. Clearly state the input and output of your model. Be explicit: “The input to our algorithm is an image, voltage of 10 sensors, etc.. We then use a CNN, LSTM, etc. to output a predicted age, decide which motor should be accelerated, etc..”
4. **Related work.** You should find existing papers, and discuss the ones that are closely related to your work. Talk about their strengths and weaknesses, as well as their similarities/differences to your work. In your opinion, which approaches were clever/good? What is the state-of-the-art?
5. **The approach you took to solve the problem.** Describe your learning algorithms and proposed models. Make sure to include relevant mathematical notation. For example, you can briefly include the loss formula you used to train your neural network. Provide a short description of how training is done in your project. We are looking for your understanding of how your neural network works.

6. **Dataset.** describe your dataset: how many training/validation/test examples do you have? Did you consider any preprocessing (eg., normalization)? If needed, include references about the datasets. Depending on available space, show some examples from your dataset.

7. **Thorough analysis of your work.** You should give details about what (hyper)parameters you chose (eg., the learning rate, the batch size and etc.) and how you chose them. Make sure you have explained the evaluation metrics (eg., classification accuracy, Inception score, etc.). Provide equations for the metrics if necessary. For results, you want to have a mixture of tables and plots. If possible, include both quantitative and qualitative results. Include visualizations of results, examples of where your algorithm failed and a discussion of why your algorithm failed or succeeded. Your plots should include legends, axis labels, and have font sizes that are crisp and readable if printed.

8. **Conclusions and future work.** Summarize your report and your findings. Why do you think your model worked better than others? Why did it fail? For future work, if you had more time, more team members, or more computational resources, what would you explore?

Grading. We employ a customized version of the Blackman criteria (named after D. R. Blackman who wrote the original document), presented in the table below, to mark your report. The report will be judged based on its clarity, the novelty of the problem, and the technical quality and significance of the work.

Mark	Comment
90-100	With appropriate adjustments, you can submit your work to a refereed conference/journal.
80-89	You'd brag about your work over the bar or at a party!
70-79	Commendable effort, but lacking some insights or understanding
60-69	Just a project.
50-59	Adequate.

Table 3: Course project - weight of each component

1.3 Presentation

You should record a 10min video (MP4 format) describing your project, your findings, the solution you considered and submit it by the end of week 12. This is a hard deadline and will not be extended.

We recommend your video to be slides with voice over. There are free software that you can use for screen capturing (eg., Screen-o-matic). PowerPoint also comes with recording features. Use the table 4 as a guideline in preparing your video. Note that this breakdown is only a suggestion, feel free to arrange according to the specifics of your work. The video needs to be self-contained: any ECE4179/5179 student should be able to understand what you did (at least at a high level) without consulting any other materials.

Remark 1. *Resist the urge to describe everything in great detail to avoid going over the time limit*

Grading. The video will be judged based on its clarity, and technical quality.

Deliverable. Aside from your proposal, report and presentation, teams must submit their code. Code should be submitted in a format that is easy to run, ie., we should only need to run a single Python script or Jupiter notebook per "experiment".

Item	Description	Duration
Title page	Start with a slide of the title of the project, and names of the group members.	10sec.
Problem statement and its significance	Describe the problem that you have tackled clearly. Has the problem been solved before? If so, how good was the solution.	110sec.
Related Work	Explain how the problem has been addressed before	1min.
Methodology	Start with a brief description of the overall approach; Possibly include a diagram of your pipeline and/or architecture. Describe the novel or most significant parts of your approach, Use diagrams whenever possible	3min
Results and Discussion	Describe how performance was evaluated, Report the most interesting and unexpected results[1], Why did you get those results? Possibly include diagrams comparing your work and others work.	3min
Conclusions	Describe what you have learned, Will you change anything if you were given another chance to do your project? What would you do if you have more time to work on the project?	1min

Table 4: Course project - Video Content

2 Default Project

As we have seen in labs, training a deep neural network from scratch requires a large amount of curated and labelled training data. In our current, rapidly expanding digital world, the amount of data being generated is mindbogglingly vast, with an estimated 2.5 quintillion bytes of data created every single day. However, curated and labelling data (in the neat and simplified form we are familiar with) is very difficult to come by - it requires human effort and often, expert knowledge. One of the Holy Grail's of machine learning would be to train with no labeled data. That is, have the algorithm learn to do what we want simply by learning and extracting patterns in the data. We are still far away from being able to do this for many ML problems but research is ongoing. One issue is that at some stage of training, in order to get your algorithm to do what you want, you need to tell it what your data "means" or what it "is". In the field of Semi-supervised learning, algorithms aim to improve model performance on tasks where labelled training data is limited by leveraging the large amounts of unlabelled data often available.

2.1 Semi-supervised learning

Semi-supervised learning can be broken into two categories: one-stage or two-stage semi-supervised learning. Two stage approaches consist of an unsupervised (or self-supervised - another ML field itself) pre-training phase (using unlabelled data only) and a fully supervised fine-tuning phase (using labelled data only). Fine-tuning the network from the pre-trained weights should result in better performance than starting with random weights, in scenarios where labelled data is limited. One-stage approaches learn from unlabelled and labelled data simultaneously.

In this research project, you are to investigate and implement a of semi-supervised learning method for image classification. We recommend focusing on two-stage methods, as they are often simpler. Note that the term "unsupervised" can be a bit misleading, and there will indeed be a supervisory signal to drive the learning. However, the supervisory signal is not provided with the data - it will need to be created from the data. Therefore, "self-supervised learning" can be a more accurate term and should be included in some of your Google searches.

2.2 Creating a Baseline

To get started you should create a baseline test-set accuracy for the dataset by training a ResNet18 model from **Randomised** weights. This baseline should then be used to compare the results from existing semi-supervised learning methods you find from your research. We have provided a skeleton code to help you start.

The self-supervised methods should be used (in some way) to pre-train the RenNet18 model from the **unlabeled** data. The pre-trained weights should then be used to as an initialisation to train on the **labelled** data. Once you compared existing methods, you must use your new knowledge of Semi-supervised\Self-supervised learning to create your own method of improving the accuracy of a classifier with unlabeled data.

2.3 Data

We will be using a semi-supervised dataset called STL10, which contains labelled images belonging to 10 object classes (airplane, bird, car, cat, deer, dog, horse, monkey, ship and truck). Additionally, the dataset contains about 100,000 unlabelled images. The unlabelled examples are from a similar, but larger distribution. For example, it may contain images of other vehicles (e.g.trains, buses etc.) or animals (bears, rabbits etc.).

The dataset is obtained via the torchvision package and we have provide code to extract the correct training splits. To simulate a scenario of very limited labelled training data, we will use only a small subset of the available labelled data (only 10%!) as the training set. The remaining data will be used as validation and test sets. You MUST use and report on the training/test/validation splits provided in the skeleton code.

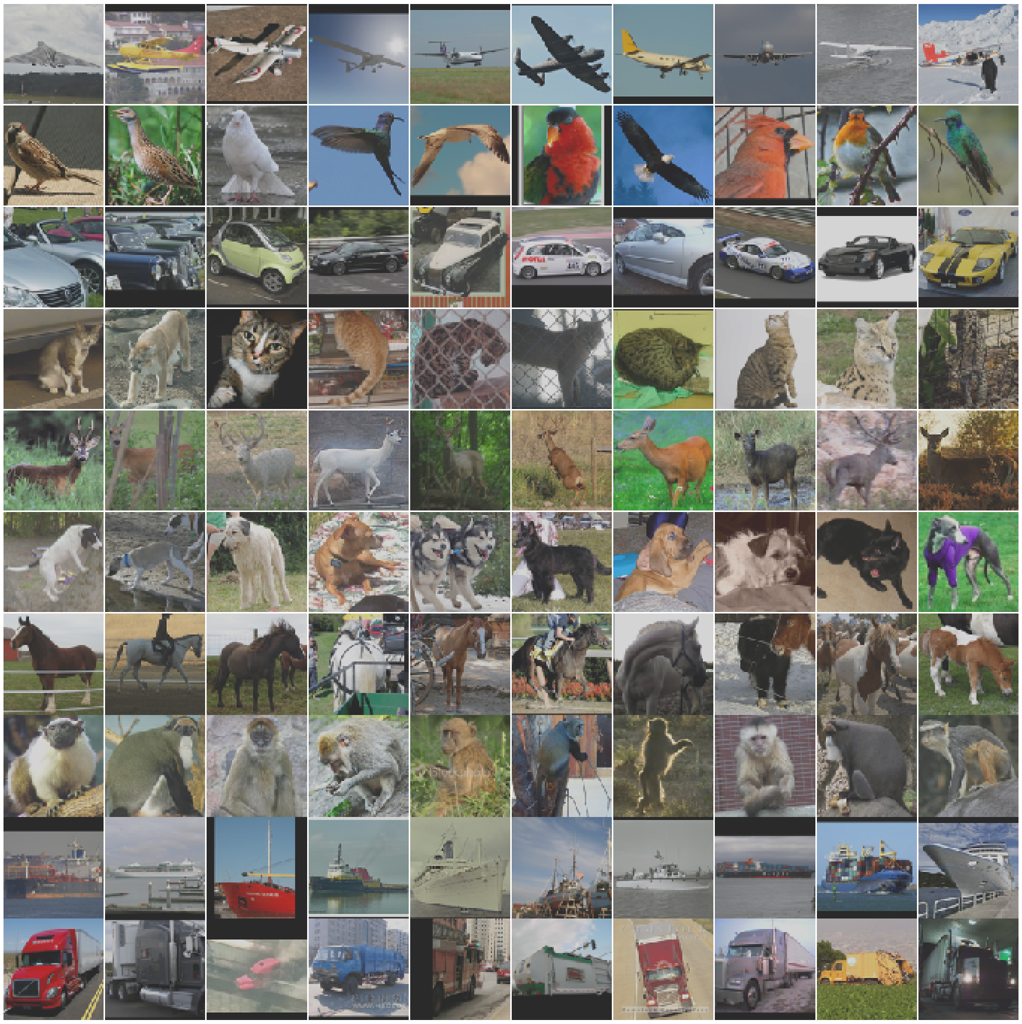


Figure 1: Example images from the STL10 datasets.

2.4 Resources

Do your own research to find useful resources, papers, ideas, blogs, videos, tutorials etc. We provide just a few below:

- Wikipedia page on semi-supervised learning (link)
- Self-supervised pre-trainig (link)
- Example of one-stage semi-supervised learning (link)
- Revisiting Self-Supervised Visual Representation Learning (link)
- How Useful is Self-Supervised Pretraining for Visual Tasks? (link)
- Selfie: Self-supervised Pretraining for Image Embedding (link)

3 Final Recommendations

1. Do not over-commit! The last 10% of a Deep Learning project is the most difficult part and you often don't know how difficult it will be until you get there. Start start slow and build up until you get to your last "grand idea" (documenting everything and collecting results as you go!).
2. Start early! Depending on the project you may need to spend weeks "fine tuning" hyper-parameters and various other specific details of your project implementation. The right set of hyper-parameters could take your algorithm from "not working at all" to "working perfectly".
3. Transfer Learning! As we've learnt/will learn in labs a good pre-trained feature extractor can give you a jump start in your training, even when your task isn't classification!
4. Have a look at existing code. Even though you must come up with your own original contribution you don't have to re-invent the wheel. Have a look at existing projects online and see what basic functions/classes/methods you can use, this includes lab code (all with proper citation of course).

For any questions/comments, please contact Mehrtash, Luke, or Himashi.