Homework #3

Deep Learning for Computer Vision

NTU, Fall 2021

110/11/23

110/12/14 (Tue.) 3:00 AM (GMT+8) due

Outline

- Problems & Grading
- Dataset
- Rules

Problems – Overview

- **Problem 1:** Image Classification with Vision Transformer (80%)

  [hw3_data/p1_data]

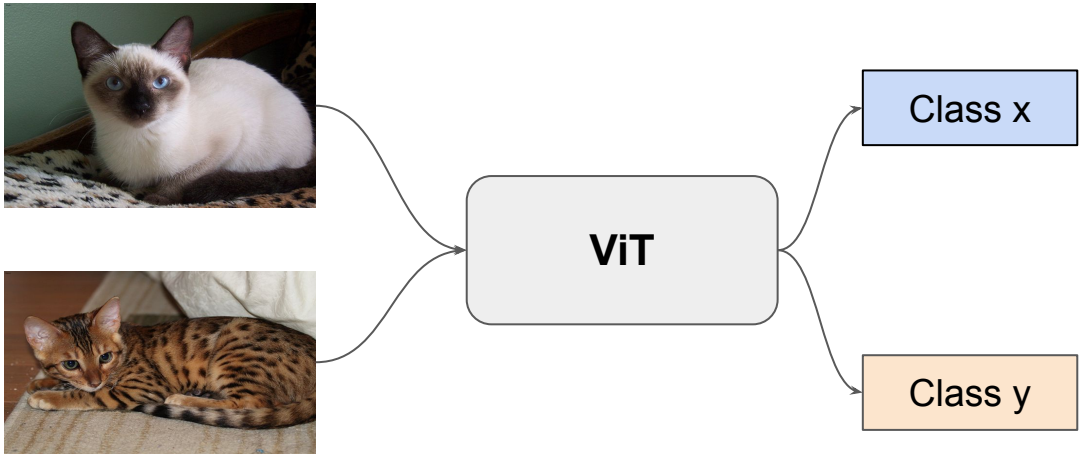
- **Problem 2:** Visualization of Attention in Image Captioning (20%)

  [hw3_data/p2_data]

Please refer to “Dataset” section for more details about p1_data/p2_data datasets.

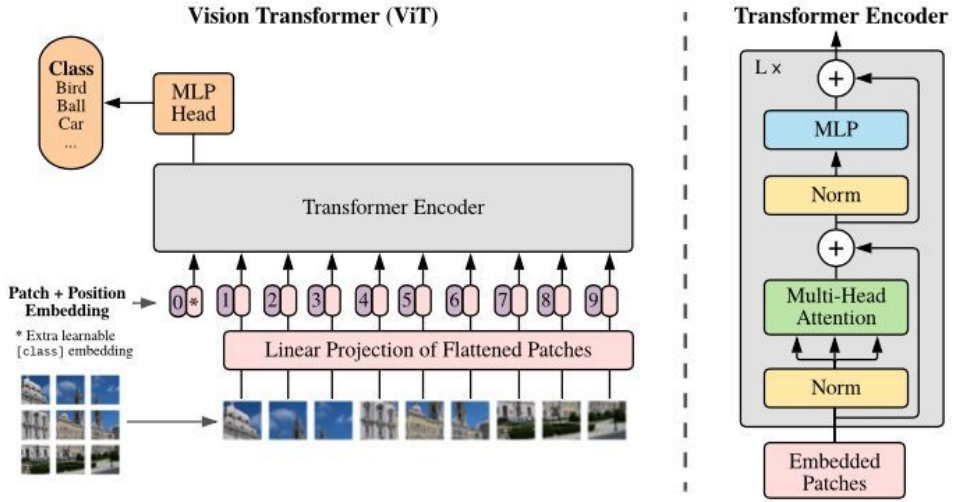
Problem 1: Image Classification with ViT

- You will need to train a **Vision Transformer** for image classification.
  - Input: RGB image
  - Output: Classification label



Problem 1: Model

- You are allowed to use any **pretrained** ViT model (e.g. ViT-Small, ViT-Base) and finetune on our dataset.



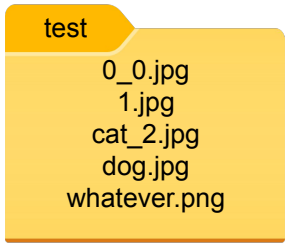
Reference: [An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale]
GitHub: [PyTorch-Pretrained-ViT], [timm]

Problem 1: Evaluation

- Evaluation metric: Accuracy

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

  - Accuracy is calculated over all test images.
- Sample Output
  - Output format: CSV file with the first row being ‘filename, label’
  - Output filename must be identical to input filename



	A	B	C
1	filename	label	
2	0_0.jpg	0	
3	1.jpg	0	
4	cat_2.jpg	0	
5	dog.jpg	0	
6	whatever.png	0	
7			

Problem 1: Grading - Baselines (30%)

- Public Baseline (15%) - 1500 validation data (p1_data/val/)
  - Simple baseline (10%) - Accuracy of 90%
  - Strong baseline (5%) - Accuracy of 94%
- Private Baseline (15%) - 2169 testing data (not available for students)
  - Simple baseline (10%)
  - Strong baseline (5%)

Problem 1: Grading - Report (50%)

1. Report accuracy of your model on the validation set. (TA will reproduce your results, error ±0.5%) (10%)
2. Visualize position embeddings of your model. (20%)
3. Visualize attention map of 3 images. (p1_data/val/26_5064.jpg, p1_data/val/29_4718.jpg, p1_data/val/31_4838.jpg) (20%)

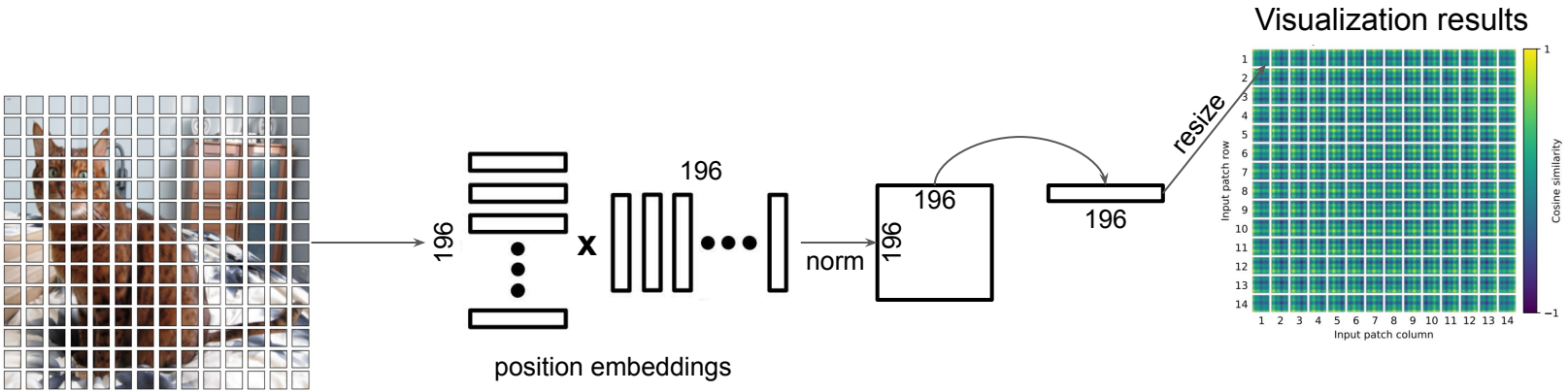
Please refer to the following slides for more detail.

Problem 1: Grading - Report (cont'd)

1. Report accuracy of your model on the validation set. (TA will reproduce your results, error ±0.5%) (10%)
    a. Discuss and analyze the results with different settings (e.g. pretrain or not, model architecture, learning rate, etc.) (8%)
    b. Clearly mark out a single final result for TAs to reproduce (2%)

Problem 1: Grading - Report (cont'd)

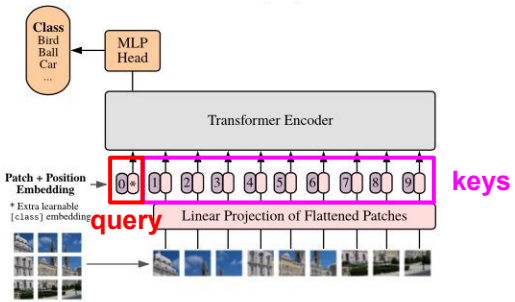
2. Visualize position embeddings (20%)
   a. Visualize cosine similarities from all positional embeddings (15%)
   b. Discuss or analyze the visualization results (5%)



Reference: [An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale]

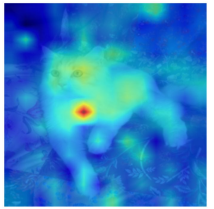
Problem 1: Grading - Report (cont'd)

3. Visualize attention map of 3 images (p1_data/val/26_5064.jpg, p1_data/val/29_4718.jpg, p1_data/val/31_4838.jpg) (20%)
   a. Visualize the attention map between the **[class] token (as query vector)** and **all patches (as key vectors)** from **the LAST multi-head attention layer**. Note that you have to average the attention weights across all heads (15%)
   b. Discuss or analyze the visualization results (5%)



Image

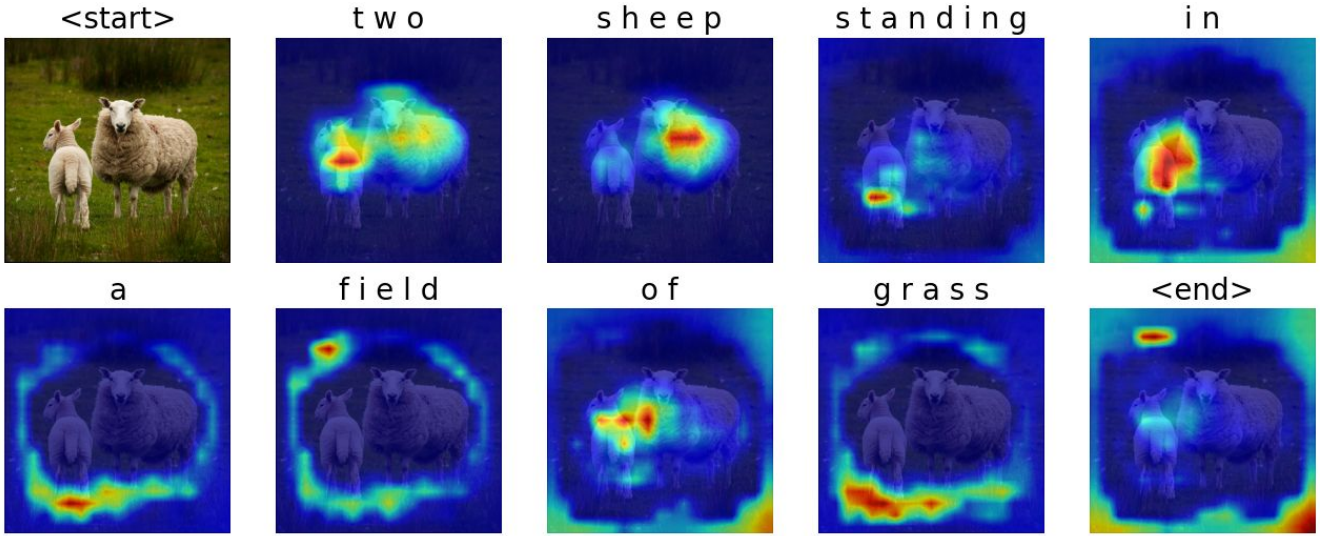
Visualization results



Reference: [An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale]

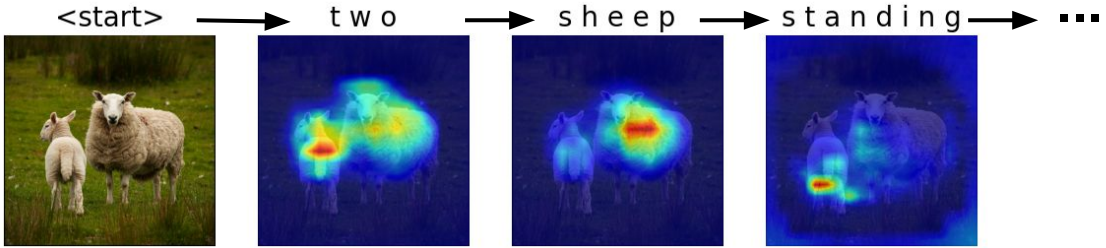
Problem 2: Visualization in Image Captioning

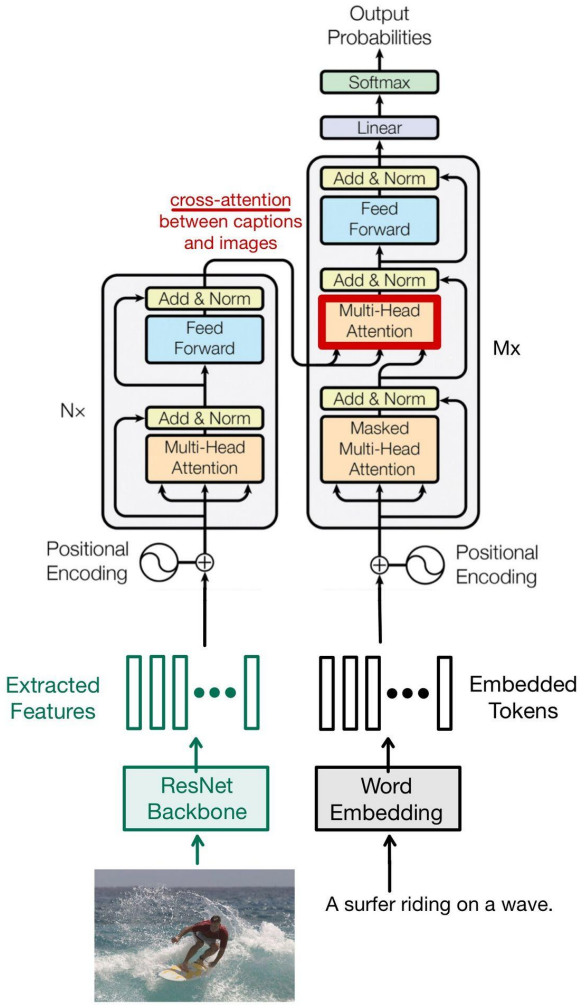
- **Transformer-based** architectures have shown great success in image captioning.
- You have to analyze the transformer decoder in image captioning by visualizing the **cross-attention** between images and generated captions.



Problem 2: Models and Settings

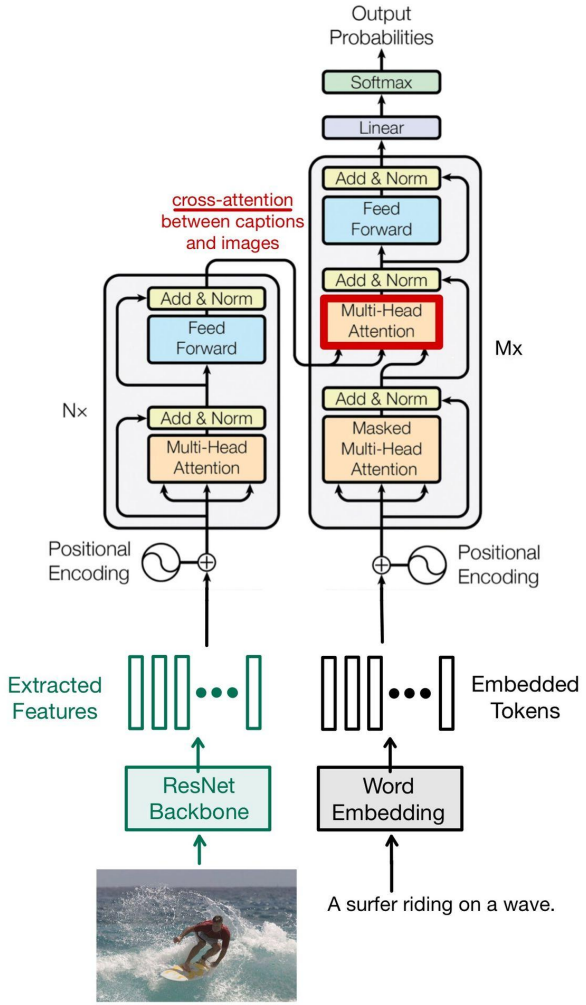
- You are allowed to use any pretrained **Transformer** architecture for image captioning.
- Given an input image, your model would be able to generate a corresponding caption sequentially, and you have to visualize the cross-attention between the image patches and each predicted word in your own caption.





Problem 2: Models and Settings

- The cross-attention weights you have to visualize are in the **decoder** part of the Transformer architecture.
  - The cross-attention weights in the last decoder layer (or the second to last) have better visulization results.
  - You have to trace the encoder and the feature extraction backbone to understand the mapping between image patches and the attention weights.
  - We recommend you to visualize attention maps for each word after the decoder generates the end-of-sequence token (EOS) (i.e., all previously predicted word token are taken as inputs of the decoder at this time.) so that you are able to visualize the cross attention maps of all words at the same time.
- **Practically, you are recommended to study the model architecture in the following Github reference.** That is, you are allowed to load its pretrained model (choose either v1, v2, or v3), and **modify** the source code for visualization and analysis.
  - https://github.com/saahiluppal/catr



Problem 2: Evaluation - Output Samples

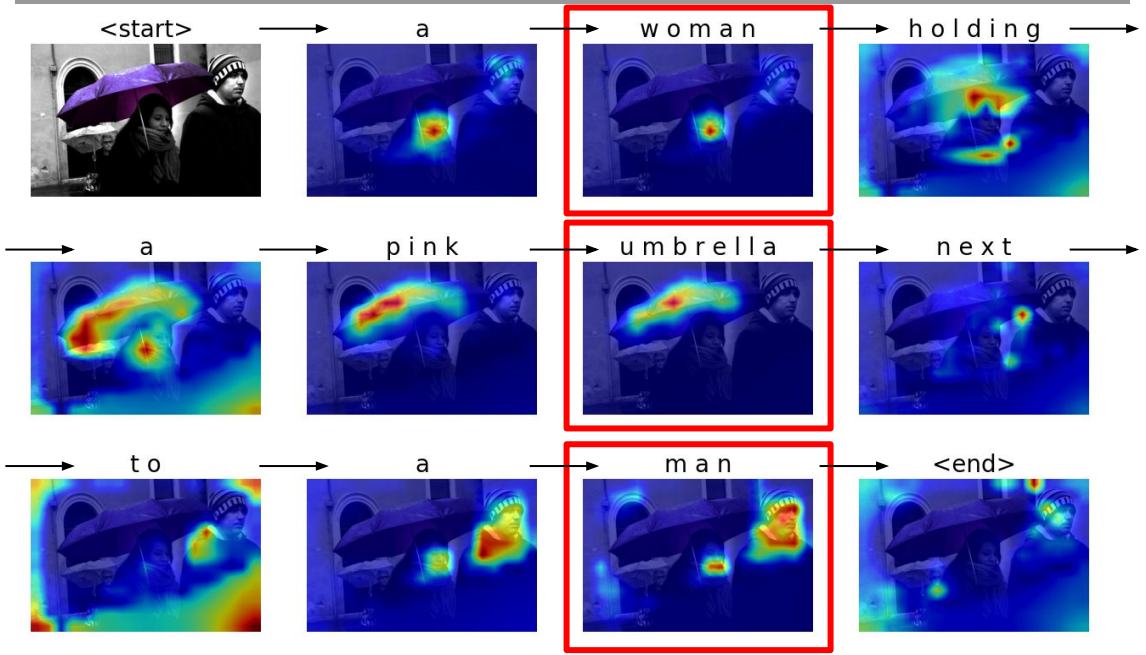
- Your visualization outputs (a PNG file) must contain the **predicted caption** & **attention maps**

Ground-Truth Captions for “umbrella.jpg”

URL

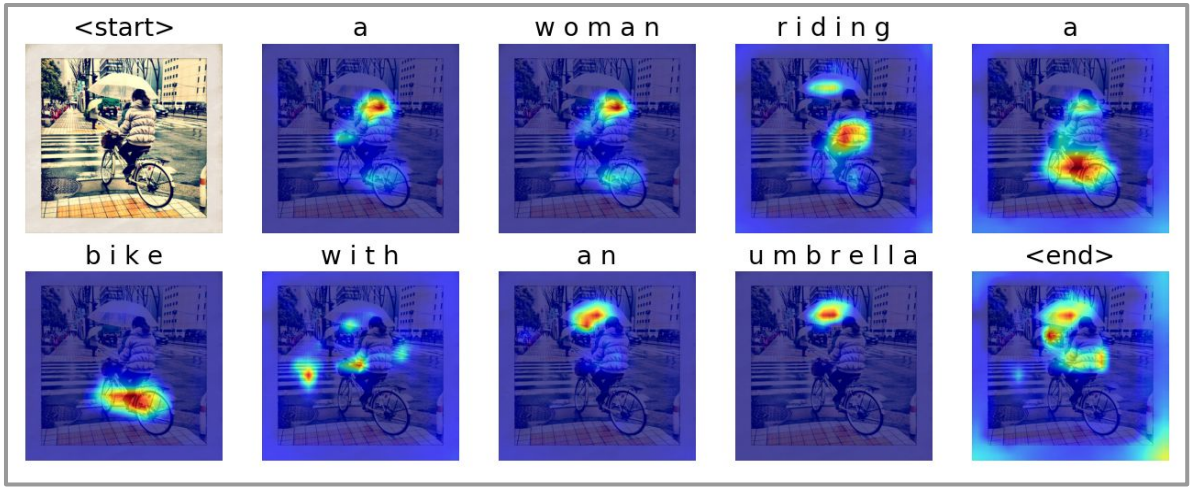
http://cocodataset.org/#explore?id=114674
http://farm9.staticflickr.com/8192/8416982488_3c40f539e1_z.jpg
a lady holding a bright purple umbrella among others.
a man and woman standing next to each other with the woman holding an umbrella.
a woman walking next to a man while holding onto a purple umbella
a man and a woman walking down a street under an umbrella.
a woman with a purple umbrella walking by a man.

My Predicted Caption and Visualization



Problem 2: Grading - Results (20%)

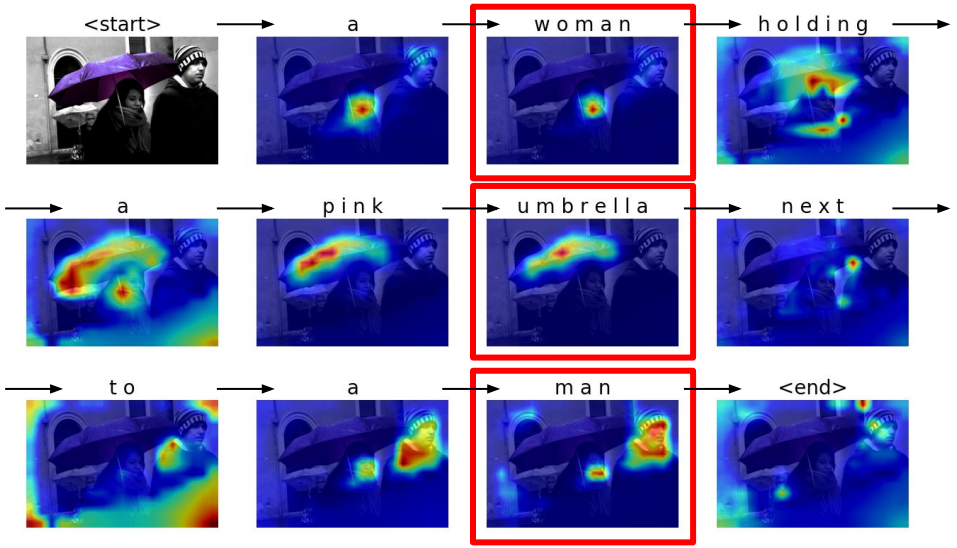
1. For the five test images, please visualize the **predicted caption** and the corresponding series of **attention maps** in a single PNG output. TA will reproduce your visualization results with your bash script. (10%)
   a. Save the **five** visualization results (PNG images) in the specified folder directory.
   b. Name your output PNG images as follows (same as the input filename):
      - bike.png
      - girl.png
      - sheep.png
      - ski.png
      - umbrella.png



An example for bike.png

Problem 2: Grading - Report (20%)

2. Choose **one** test image and show its visualization result in your report. (10%)
   a. Analyze the predicted caption and the attention maps for each word. Is the caption reasonable? Does the attended region reflect the corresponding word in the caption?
   b. Discuss what you have learned or what difficulties you have encountered in this problem.



Outline

- Problems & Grading
- Dataset
- Rules

Problem 1: Dataset

- The dataset consists of **5180** pet images in **37** classes
- We split the dataset into
  - p1_data/train/
    - **3680** images
    - Images are named ‘{class_label}_{image_id}.jpg
  - p1_data/val/
    - **1500** images
    - Naming rules are the same as p1_data/train/
    - Note that you **CANNOT** use validation data to train your model

Problem 2: Dataset - MS COCO 2017

- COCO (Common Objects in Context) is a large-scale object detection, segmentation, and captioning dataset. Each image has five captions.
- We provide **five** images (from MS COCO 2017 Explore) for captioning, and TA will reproduce your visualization results (predicted captions and attention maps) only with these five images.
- These five test images and as below:
  - **p2_data/images/**
    - **bike.jpg**
    - **girl.jpg**
    - **sheep.jpg**
    - **ski.jpg**
    - **umbrella.jpg**





- You can find the five ground-truth captions for each image in **p2_data/caption.txt**.

Outline

- Problems & Grading
- Dataset
- Rules

Submission

- Click the following link and sign in to your GitHub account to get your submission repository

  https://classroom.github.com/a/FSNiwFEi

- By default, we will only grade your last submission before the deadline (NOT your last submission). Please e-mail the TAs if you'd like to submit another version of your repository, and let us know which commit to grade.

Submission

- Your GitHub repository should include the following files
  - hw3_<studentID>.pdf
  - hw3_1.sh
  - hw3_2.sh
  - Python files (testing code & visualization code)
  - Model files (can be loaded by your python code)
- **Do NOT upload your dataset.**
- If any of the file format is wrong, you will get zero point.

Trained Model

- If your model is larger than GitHub's maximum capacity (100MB), you can upload your model to another cloud service (e.g., Dropbox). However, your script file should be able to download the model automatically. (Dropbox tutorial)
- **Do NOT** delete your trained model before TAs disclose your homework score. **Do NOT** delete your trained model before you make sure that your score is correct.
- Use the **wget** command in your script to download you model files. **Do NOT** use the curl command.
- Note that you **should NOT hard code any path** in your file or script except for the path of your trained model.

Bash Script - Problem 1

- TA will run your code as shown below
  - bash hw3_1.sh $1 $2
    - $1: path to the **folder** containing test images (images are named xxxx.jpg, where xxxx could be any string)
    - $2: path of the output csv **file** (e.g. output/pred.csv)
- Please follow the naming rules in p.6
- Note that you should **NOT** hard code any path in your file or script.
- Your testing code have to be finished in **10 mins.**

Bash Script - Problem 2

- TA will run your code as shown below
  - bash hw3_2.sh $1 $2
    - $1: path to the **folder** containing test images (e.g. captioning/images/)
    - $2: path to the **folder** for your visualization outputs (i.e. five PNG files) (e.g. hw3/p2_output/)
- The five output PNG files **must** follow the naming rules (same filename as the input test images).
- Note that you should **NOT** hard code any path in your file or script.
- Your testing code have to be finished in **10 mins.**

Bash Script (con’d)

- You must **not** use commands such as **rm**, **sudo**, **CUDA_VISIBLE_DEVICES**, **cp**, **mv**, **mkdir**, **cd**, **pip** or other commands to change the Linux environment.
- In your submitted script, please use the command **python3** to execute your testing python files.
  - For example: python3 test.py $1 $2
- We will execute you code on **Linux** system, so try to make sure you code can be executed on Linux system before submitting your homework.

Packages and Reminders

- Python==3.8
- Please refer to **requirements.txt** on your hw3 GitHub repository.
- **Do not** use **imshow()** or **show()** in your code or your code will crash.
- Use **os.path.join** to deal with path as often as possible.
- If you train on GPU ids other than 0, remember to deal with the "**map location**" issue when you load model. (More details about this issue, please refer to https://github.com/pytorch/pytorch/issues/15541)

Outline

- Task description & Implementation details
  - Problem 1: Image Classification using Vision Transformer
  - Problem 2: Visualization of ?
- Submission
- Homework Policy

Deadline and Academic Honesty

- Deadline: **110/12/14 (Tue.) 3:00 AM (GMT+8)**
- Late policy : Up to 3 free late days in a semester (depends on your hw0 result). After that, late homework will be deducted 30% each day.
- Taking any unfair advantages over other class members (or letting anyone do so) is strictly prohibited. Violating university policy would result in F for this course.
- Students are encouraged to discuss the homework assignments, but you must complete the assignment by yourself. TA will compare the similarity of everyone's homework. Any form of cheating or plagiarism will not be tolerated, which will also result in F for students with such misconduct.

Penalty

- If we cannot execute your code, TAs will give you a chance to make minor modifications to your code. After you modify your code,
  - If we can execute your code, you will still receive a 30% penalty in your model performance score.
  - If we still cannot execute your code, no point will be given.

Reminder

- Please start working on this homework as early as possible.
- The training may take a few hours on a GPU or days on CPUs.
- Please read and follow the HW rules carefully.
- If not sure, please ask your TAs!

How to Find Help

- Google!
- Use TA hours (please check course website for time/location).
- Post your question under HW3 discussion section on NTU COOL.
- Contact TAs by e-mail: ntudlcv@gmail.com.

DOs and DONTs for the TAs (& Instructor)

- Do NOT send private messages to TAs via Facebook.
- TAs are happy to help, but they are not your tutors 24/7.
- TAs will NOT debug for you, including addressing coding, environmental, library dependency problems.
- TAs do NOT answer questions not related to the course.
- If you cannot make the TA hours, please email the TAs to schedule an appointment instead of stopping by the lab directly.