UMKC SSE
CS5567 Deep Learning Final Project -- SP 24

IMPORTANT: (please read and follow all steps!)

Instructions:

1. **Take this step-by-step.**
2. **Don't get overwhelmed.**
3. **Remember that we're here to support you, so please ASK for assistance!**

- FILES:
  - Download "Final.zip" – it contains (.ipynb) files.
  - Where appropriate load the scripts given for each experiment in your **Colab** or **local python** workspace. If you need help setting up a local workspace, we'll be happy to assist!
  - Most of the data files you'll need will be loaded as part of the scripts, but some will require a bit of integration (many examples and links are provided).

- Instructions:
  - There are 2 Parts **A & B**
  - The ***Experiments*** table (below) includes guidelines/instructions.

- Results:
  - **Save** a **record** of the **values** you achieved along with EACH **experimental configuration** you tested.
  - Both the TRAIN and TEST metrics are important to the assessment of each task. There are a variety of ways these data may be displayed and reported, but always remember that a **VISUAL** representation {plot} will be **superior** to a simple numerical representation {table}.
  - You will provide **1-2** pages of your **formal FINAL REPORT** on your findings from the experiments in PARTS **A** and **B** using appropriate terms, and references* (where applicable).
    *Reminder: All generated material **must** provide reference to the source model.

- **Final Report:** Generate an explanation of your efforts and findings as well as the responses to the questions requested under "Discussion" at the end of the document.
  This will be in the **from** of 2 documents:
  - **ONE report document (.docx or .pdf)**
    - **Notes, observations, and Responses to Discussion** – your observations when implementing the requested processes.
  - **ONE report file (.xlsx or .csv)**
    - Save the output of your training & testing processes as part of this file.
    - **Use will also need to use this data to generate:**
      - **Figures, tables, or graphs** for observations made in the **discussion**.

Your **DISCUSSION VIDEO PRESENTATION will be turned in via a separate link on canvas. This takes the place of a "final" examination.** The video will be recorded/uploaded via **Panopto**

This video highlight your RESULTS, *so be sure to* **make notes while completing this activity so you can refer to them later.**

Experiments

Part A: Utilizing Pre-Trained Feature Layers in Adjacent Tasks

Experiment 1	**End-to-End Classification (Color Images)**
Data Set	CIFAR-10
CNN Topology	CUSTOM – **Image Classification (10 class)** ❖ Utilize knowledge gained thus far to determine which topology and hyperparameters are likely to provide reliable performance for this task. ❖ Grid search is **unlikely** to be ideal or necessary, so a more "informed" stochastic approach (a good *hypothesis* is an **educated guess**!)
Layers & Elements	Utilize the (.ipynb) in conjunction with the Keras API documentation: https://keras.io/api/ ➢ Generate a classification model for color image data on the referenced dataset. ➢ Train your method using your preferred optimization functions. ➢ Update topology as needed (within limits) *[max of 5-7 configurations]*
Modifications	Make whatever hyperparameter changes you need to arrive at a reasonable level of performance [3-5% ≤ industry standard] in classification for the listed Data Set.
Reporting	Add the results of your training &/or testing (output) to your .xlsx or .csv file ➢ You should track your model performance during training. ➢ You should report both training and testing accuracy (and validation if you plan to implement it).
Save Your Model!	https://keras.io/guides/serialization_and_saving/ You will **NEED** it for **Experiments 2 & 3**

Experiment 2	**Transfer Learning**
Data Set	CIFAR-10 / CIFAR-100 / Tiny ImageNet ← colab import example
Pre-Trained Models	❖ Included (vgg19.ipynb) Reference → VGG 19 ❖ Included (resnet.ipynb) Reference → RESNET
Parameters	Utilize the Keras API TRANSFER LEARNING documentation: https://keras.io/guides/transfer_learning/ ➢ Look first at the section under the heading → "The typical transfer-learning workflow" ➢ Load the model and weights ***for each*** Pre-trained model above. ➢ Add the ***classification*** portion ***for each*** dataset referenced above. ➢ **Train** your classification layers using your preferred topology and optimization functions.
Modifications	You should "freeze" the model after importing e.g. → base_model.trainable = False
PROCESS	For Each Pre-Trained Model → "Train the top layer" (search term in reference) [add the DENSE elements and build the model appropriate for each Data Set] **(2 Models x 3 Datasets = 6) + (CUSTOM x 2 Datasets = 2) = 8 total**
Reporting	Add the results of your training &/or testing (output) to your .xlsx or .csv file ➢ Track each of your model's performance during training. ➢ You **only** need to report **testing** accuracy

Experiments

Experiment 3	**Fine-tuning**
Data Set	Tiny ImageNet ← colab import example
Process	**[no script supplied]** **Tiny ImageNet** size is 64x64 so you'll need to use the resizing function Utilize the Keras API documentation for *FINE-TUNING* Refer to the materials under section → ❖ "Do a round of fine-tuning of the entire model" (search term in reference)
Model	❖ Import your previous custom model from Experiment 1
Layers & Elements	You will need to adjust the output layer size as the number of classes (200) in this set are much larger than CIFAR-10 (10)
Modifications	**Unfreeze** the model and make sure to call ***build*** again once you've set value: base_model.trainable = True
Reporting	Add the results of your training &/or testing (output) to your .xlsx or .csv file ➢ You should track your model performance during training. ➢ You should report the final training and testing accuracy.
Save Your Model!	https://keras.io/guides/serialization_and_saving/ You will **NEED** it if you plan to do **Experiments 4**
Note:	The **goal** will be to determine how well your model adapted to the new task, a task that is much more complex than it's original goals.

Experiment 4	**Memory Loss or Functional Gain**
Model	❖ Use the "fine-tuned" version of your custom model from Experiment 3
Explanation	You should now attempt to determine if the fine-tuned version of your model **retained** it's previous level of performance, or if it **gained** any feature extraction/representation power. You can test your model again using test samples from CIFAR-10.
Reporting	Add the results of your training &/or testing (output) to your .xlsx or .csv file ➢ You should report the final training and testing accuracy.

Experiments

Part B – Segmentation and Style Transfer

Experiment 1	**Localization [ROI]**
Data Set	**https://public.roboflow.com/object-detection** Look for the file format → "**TensorFlow Object Detection CSV**"
Modifications	Get a new dataset from the link above – (or you can find any object detection dataset elsewhere) download the dataset ➢ If you want to try your hand at some data format modification - download the dataset in the "**yolov8**" [or any other compatible data format].
Process	Utilize the supplied (yolo_v8.ipynb) ❖ Modify the existing code from the notebook and **replicate** the process and **results** on your chosen dataset.
Reporting	Add the results of your training &/or testing (output) to your .xlsx or .csv file ➢ Save the images generated from your experiments ➢ Save the metrics provided for final performance (**IoU**, etc.)

Experiment 2	**SegFormer Segmentation**
Data Set	Included
Process	Utilize the (Fine_tune_SegFormer.ipynb) Follow the steps indicated in the script to load and refine the model.
Modifications	Select 3 new sample images from the dataset (**at random**) and assess the performance of the task optimized network on those images.
Reporting	Add the results of your training &/or testing (output) to your .xlsx or .csv file ➢ Save the images generated from your experiments ➢ Save the metrics provided for final performance (**IoU**, etc.)

Contest!

Generative Models	**Working with Style Transfer**
Data Set	Your own Images!
Process	https://huggingface.co/spaces/svjack/Super-Resolution-Neural-Style-Transfer https://keras.io/examples/generative/neural_style_transfer/
Objective	Select 10 images These images can come from **wherever** you please. ➢ Assure these images are appropriate in both content and theme. ❖ Select an artistic style to transfer to the images. This can be another photo, a famous artist, a tiled pattern, or texture. ❖ You can pick more than one style if you like, but indicate the source for each image.
Reporting	Submit your images (**indicate your 2 favorites**) as part of your report. Myself and the TA's will nominate their favorites, and the first, second, and third place students will receive a reward (and 10 bonus points) for their efforts.

Experiments

Discussion:

1. Compare the results of your experiments for Part A.
    a. Display and discuss the selected classification layer **training performance**.
    b. Display and discuss the results of the **test performance** for each experiment.
    c. Obviously, there's quite a bit to do here, and you can't solve it all so -- what would you change if you had time?
    d. Which model do you think provided the better set of features for training between the VGG and RESNET? How did you come to this conclusion?

2. Compare the results of your experiments for Part B.
    a. Display and discuss the results of the image segmentation using the YOLO model.
    b. Display and discuss the results of the image segmentation using the SegFormer.
    c. Which model do you feel provides the best structural understanding of geometric structures?
    d. Did you observe any issues with overlapping objects?
    e. What type of images were best for use with the Style Transfer models? Discuss both the input and style images. What do you think this says about the utility of these methods?

3. Discuss in a few sentences your observations with each of the experiments. How do the things you observed relate to the things we've covered in this course? **Do not** simply include the **definition** of these topics – instead speak about how you feel your efforts relate to the outcomes.

4. Deep Learning has provided us with several amazing advancements over the past few years, take a few minutes to consider what might be next in the field.
    a. Which of the major topics that we've discussed do you think you will spend more time on in the future?
    b. Do you think Deep Learning is an example of emerging intelligence in machines?
    c. Provide one constructive comment/suggestion/idea which can be made to assist students in this course in future semesters.