Project Abstract Examples

Visual Counting

Visual counting is a problem in computer vision, which requires accurately counting object instances in scenarios. Visual counting has a substantial advantage over object detection and even object classification: Data collection and annotation. Compared to object detection, object counting does not required human-labeled accurate tight bounding boxes around the objects. Compared to object classification, object counting does not require clean dataset that contains one object per sample. Thus we come to the main idea of this paper: we take use of this advantage to train visual counting first, and then transfer the model to other tasks. The experiment results validate our idea and demonstrate a possible way to solve core recognition tasks like detection and classification, with relatively easily collected dataset and little-effort annotations.

Towards CPU Real-Time Object Detector

In recent years, the trade-off between accuracy and speed in detection has been widely researched. As many single-stage detectors have achieved remarkable performance on both sides, the performance of CPU real-time detectors is still far from satisfaction. In this project, we design a CPU real-time detector based on SSD baseline. First, we train a teacher network which has high accuracy based on SSD and FPN. Then we do model compression using knowledge distillation with MobileNet+SSD as student network, which is much faster while maintaining the accuracy. Experiment results demonstrate that our proposed methods could increase detection results of the tiny student network especially on hard classes like tiny or occluded objects.

Visual Question Answering

Visual Question Answering (VQA) is a system that can answer natural language questions about any image. This is a multimodal research problem. We propose exploring different approaches to the existing baseline bottom-up top-down attention approach to the VQA problem by introducing 3 new models that are variations to the original model. The first uses image captions as context in the top-down approach along with the question. The second gives the image features as input to the GRU at every time step rather than combining it with the output of the GRU only at the end. The third model uses transformers instead of pretrained GloVe embeddings to create the word embeddings for the question. We also explore combinations of different attention mechanism with the bottom-up top-down mechanism and evaluate performance of each on a subset of MS-COCO dataset.

Localized Video Style Transfer

The project we chose is localized video style transfer. Style transfer is the technique of recomposing images or videos in the given style. In some application scenarios such as video effects and rendering, users may only want background to be style transferred while the rest of the scene (e.g., foreground objects) remains the original style, in order to achieve certain aesthetic goals. For example, for a travel video, the user might want an effect of people walking in an oil paint style scenery. Therefore, instead of traditional full range style transfer, our project is focused on style transfer the background part of a video, while keeping the foreground objects (e.g., cars, pedestrians, and boats) in the video as their original appearances.

The inputs are an original video and a target style image, and the output is the localized style transferred video. Considering the aesthetic traits of this project, we evaluate the results from two perspectives, namely video quality evaluation (e.g., smoothing) and video good looking based on user study.

Given a video and a style image, the complete pipeline of this project is summarized as follows:

- Extract all frames of the video;
- Apply video style transfer on full range for the video;
- Generate all frames of the style transferred video;
- Do object detection and tracking; It can be done in either of the two ways:

  –Apply Mask R-CNN on each frame (to accelerate computation, it is better to operate this step on GPU's);

  –Occasionally call Mask R-CNN. Between the adjacent neural network calls, apply Lucas-Kanade to track the existing masks in the new frame.
- For each frame, do binary morphology for all foreground object masks;
- Mask out background and generate localized style transfer frames;
- Generate localized style transfer video.

Generating Adversarial Input Perturbations using Feedforward Network

In today's society where public facing AI is becoming ever more prevalent, it's increasingly necessary to consider the security and reliability of the algorithms we use. In this paper, we will tackle the problem of AI security by targeting a neural network, adversarially generating small data adjustments designed to cause huge changes in network predictions. To this end, we will train our own neural network that will learn small changes in input data to cause the largest perturbance in the performance of a target neural network. These changes should be hard to detect with human inspection. Armed with such adjustments, an adversary could manipulate the output of a neural network with arbitrary valid input data. To get a broad sense of how useful such an attack will be, we will apply it in two dramatically different contexts: a LSTM model, used for natural language classification, and a ENet model, used for semantic segmentation. These two models are both ground-breaking advances in their respective tasks, so a hypothetical attacker model could have incredible potential countering real-world models.