21050111026 - Faruk KAPLAN
6 June, 2024

CENG 428: Neural Networks
Take Home Exam Q2

1 2nd Question

1.1 System Diagram

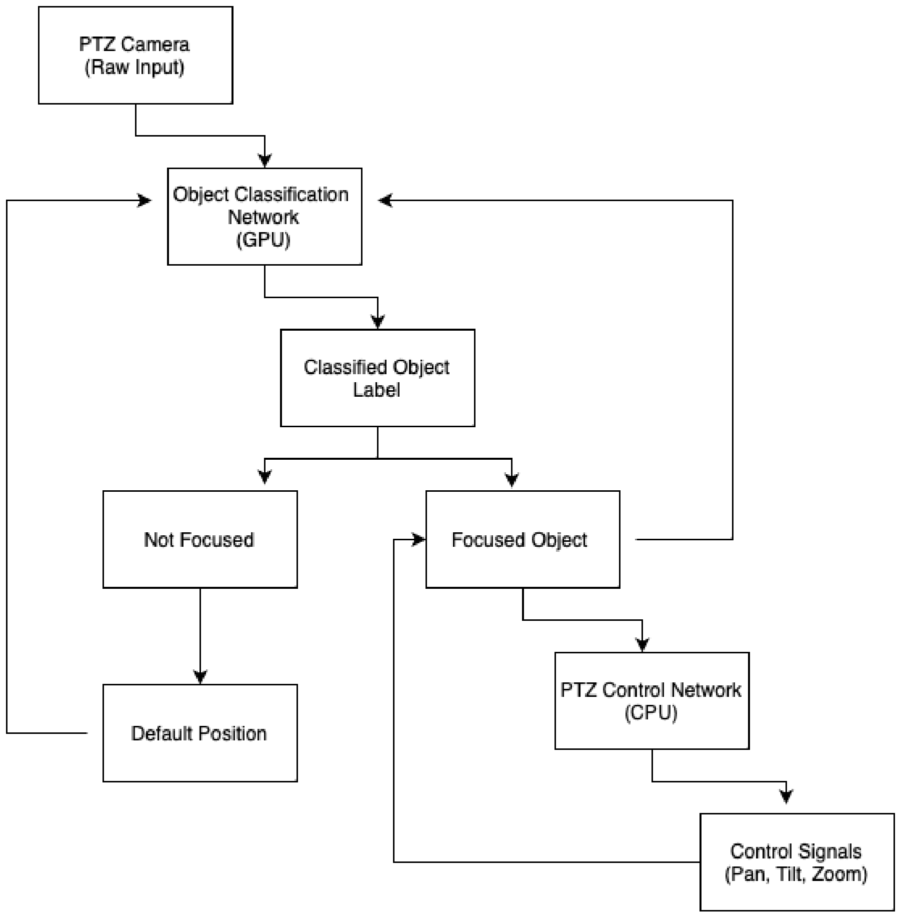
PTZ Camera (Raw Input)
Object Classification Network (GPU)
Classified Object Label
Not Focused
Focused Object
Default Position
PTZ Control Network (CPU)
Control Signals (Pan, Tilt, Zoom)

Figure 1: System Diagram

1.2 Details of NN Architecture

Object Classification Network: We have a pre-trained classification network that firstly takes input from default camera position. We can use MobileNetV2 as a pre-trained model in here. After classify all objects in the view, we label these objects, whether it is interested object or not.

- Input Layer: Takes the raw video frame.
- Convolutional Layers: Extract features from the input image.
- Residual Blocks: Enable deep feature extraction with skip connections.
- Global Average Pooling: Reduce the spatial dimensions.
- Fully Connected Layers: Output class probabilities.

- Binary Cross Entropy classifies objects

Focused Object: After the first part, we focus or don't focus to interested objects. If we don't focus something, we keep proceeding the camera input. If we focus on object, we'll go to PTZ control network.

PTZ Control Network: In here we have CNN and RNN components for adjusting camera position.

- CNN
  - Input Layer: Takes the raw video frame.
  - Convolutional Layers: Extract spatial features from the input image.
  - Global Average Pooling: Reduce spatial dimensions while retaining important features.
  - Reshape Layer: Prepare features for input to the RNN.
- Recurrent Neural Network (LSTM)
  - LSTM Cells: Process sequential feature vectors from the CNN.
  - Fully Connected Layers: Output control signals (pan, tilt, zoom) based on LSTM outputs.

Control Signals: Adjust camera position according to PTZ control network output. Also keep checking focused object for that maybe it changes its position

1.3 Obtaining Training Data

In object classify part, we can collect images containing various objects of interest in different environments. This can include datasets like COCO or ImageNet. And annotate the collected images with corresponding object labels to create a ground truth dataset for training the classification network.

In PTZ control part, we can record video streams from PTZ cameras capturing real-world scenes with objects of interest. Also we log the PTZ camera's position and zoom settings alongside the video frames to create training samples.

1.4 Training, CPU-GPU, Optimizers, Parameters

Firstly we fetch pre-trained model and pass forward with our images.

CPU: Utilize CPUs for training the RNN component of the control network due to its sequential nature and lower computational requirements compared to CNNs.

GPU: Use GPUs for training the CNN component of the control network, as it involves intensive operations such as convolution and pooling. Also we can use GPU in classification part, due to its intensive computational nature, especially during convolutional operations.

Optimizer: We can use the Adam optimizer for classification and control networks to benefit from its adaptive learning rate capabilities.

Parameters: We can initialize learning rate small value, like 0.001. We'll use minibatch gd and train in GPU, so we can set batch size between 16 and 128.

1.5 Control of the PTZ Camera

Firstly we know focused object from classification network, we process that input (focused object is input to PTZ control network) in PTZ control network and according to that, we signal to device to pan, tilt or zoom. These signals will adjust the camera position. After that we go back to focused object, because there is a possibility to move that object.

1.6 Classifying Focused Scene

The focused scene can be classified by passing the focused video frame through the Object Classification Network (OCN), which utilizes a convolutional neural network (CNN) to extract features and classify the object within the frame. The OCN outputs the predicted label or category of the object, providing insight into the content of the focused scene.

1.7 Choice of Model

When we choose model, Resnet50 could be greate choice because it is more convenient for object detection tasks, but we should consider factors such as accuracy, efficiency, and computational complexity, we have a limited GPU. When we consider those factors, we can choose MobileNetV2

- Advantages:
  - **Efficiency:** We have limited resources in GPU, so MobileNetV2 is a good choice, because it is designed specifically for mobile and embedded devices
  - **Real-time Inference:** The lightweight nature of MobileNetV2 enables real-time inference on devices with modest computational capabilities.
- Disadvantages:
  - **Slightly Lower Accuracy:** MobileNetV2 is efficient but it may not match the performance of deeper architectures like ResNet50 in tasks requiring fine-grained object recognition or classification in highly complex scenes.
  - **Limited Capacity:** Due to its lightweight design, MobileNetV2 may have limited capacity to capture intricate features or patterns in the focused scene compared to larger architectures.

Also using an LSTM alongside a CNN for scene classification allows for capturing temporal dependencies and contextual understanding, enhancing accuracy in video data analysis. However, it introduces training complexity and potential overfitting concerns that require careful parameter tuning and regularization.