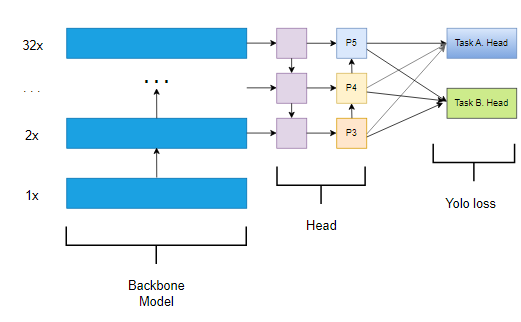
**Report CV entry test**

**[Link code](https://drive.google.com/drive/folders/1LTB_f1qSEFlsjZt7TRHaNY-29__9sHsL?usp=drive_link)**

****

### Overview of the YoloV7 Model

In this task, we utilized the YoloV7 model to simultaneously predict two classes: Head and Face. To achieve this, we employed the technique of **multitask learning**. Multitask learning allows us to modify a single neural network architecture to predict multiple tasks by adjusting the head layer of the network.

By doing so, the model can predict Head and Face detections in a single forward pass, making it efficient and effective for applications that require both detections concurrently.

### Loading the YoloV7 Model

To begin, we loaded the pre-trained YoloV7 model along with its weights. This approach is more efficient than training the model from scratch, as the pre-trained model already possesses a high level of accuracy and generalization. Using pre-trained weights allows the model to converge faster and achieve better performance on the specific task of Head and Face detection.

**Why We Chose Pre-training:**

1. **Efficiency**: Pre-trained models require less computational power and time to train.
2. **Accuracy**: The model starts with a strong baseline, which can be fine-tuned for the specific dataset.
3. **Generalization**: The model can better generalize from the pre-trained knowledge to the specific tasks of Head and Face detection.

### Loading Data and Creating the .yaml File

We downloaded the dataset from the provided source, which includes labeled data for Head and Face detection. The dataset is structured with separate directories for training, validation, and testing.

To train the YoloV7 model on this dataset, we created a .yaml configuration file. This file is essential as it informs the YoloV7 model about the dataset structure and the number of classes.

**Structure of the YAML file:**

* **Train path**: Path to the training data.
* **Validation path**: Path to the validation data.
* **Test path**: Path to the test data.
* **Number of classes (**nc**)**: Set to 2, as we have two classes: Head and Face.
* **Class names**: [“head”, “face”]

**Example YAML file:**

**+ train: path/to/train/set**

**+ val: path/to/val/set**

**+ test: path/to/test/set**

**+ nc: 2**

**+ names: [“head”, “face”]**

### Modifying the YoloV7 Model for Multitask Learning

The original YoloV7 model has a single head for object detection. In this project, we modified the architecture to have two separate heads: one for Head detection and another for Face detection.

**Modifications:**

* **Task A Head**: Designed for predicting Head detection.
* **Task B Head**: Designed for predicting Face detection.

The reason for adding two heads is to allow the model to specialize in detecting both classes with the same architecture. Each head is tailored to focus on its respective task, which improves the overall accuracy and efficiency of the model.

### Training the Model with the SCUT-HEADFACE Dataset

The model was trained using the scut\_headface dataset with the following parameters:

* **Epochs**: 100 epochs were set to ensure thorough training.
* **Batch size**: A batch size of 8 was chosen to balance memory usage and training speed.
* **Pre-trained weights**: The training started with the pre-trained YoloV7 weights for better performance.

### Inference Pipeline for Testing Images

After training, we tested the model using the inference pipeline. This pipeline processes test images and outputs the predicted bounding boxes for Heads and Faces.

|  |  |
| --- | --- |
| **Train with 38 epoch 8 batch** | |
| **Best predict:** | **Bad Predict:** |
| **download** | **download** |
|  |  |
| **Train 10epoch batch 10** | |
| **Best:** | **Bad:** |
| **download (2)** | **download** |