

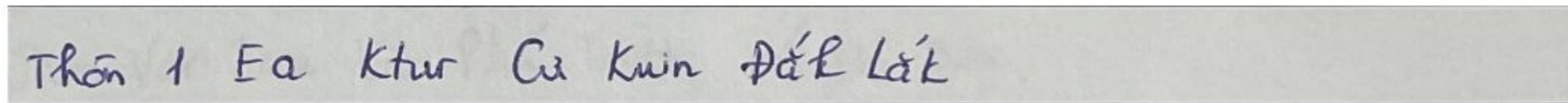
# Track2-Kalapa



# Track2-Kalapa

## Problem statement:

- Trong bài toán này, các đội thi sẽ xây dựng một model OCR light-weight giải quyết bài toán OCR chữ viết tay tiếng Việt với ngữ liệu là địa chỉ Việt Nam.



**Label:** *Thôn 1 Ea Ktur Cư Kuin Đắk Lắk*

Input & Output

# Track2-Kalapa

## Limitation

- Model size:  $\leq 50\text{MB}$
- Thời gian inference: 2s
- Môi trường inference không có kết nối internet
- Cấu hình máy inference:
  - CPU: Intel(R) Core(TM) i9-10900K CPU @ 3.70GHz
  - RAM: 64GB
- Dữ liệu: Các đội không được sử dụng các bộ dataset khác ngoài cuộc thi, trừ các bộ dataset tự sinh
- Pre-trained model: Chỉ được sử dụng các pre-trained model được công khai và được train trên tập dữ liệu ImageNet . Không được phép sử dụng các pre-trained model huấn luyện riêng cho các bài toán OCR (bất kể chữ in hay chữ viết tay).

# Track2-Kalapa

## Timeline

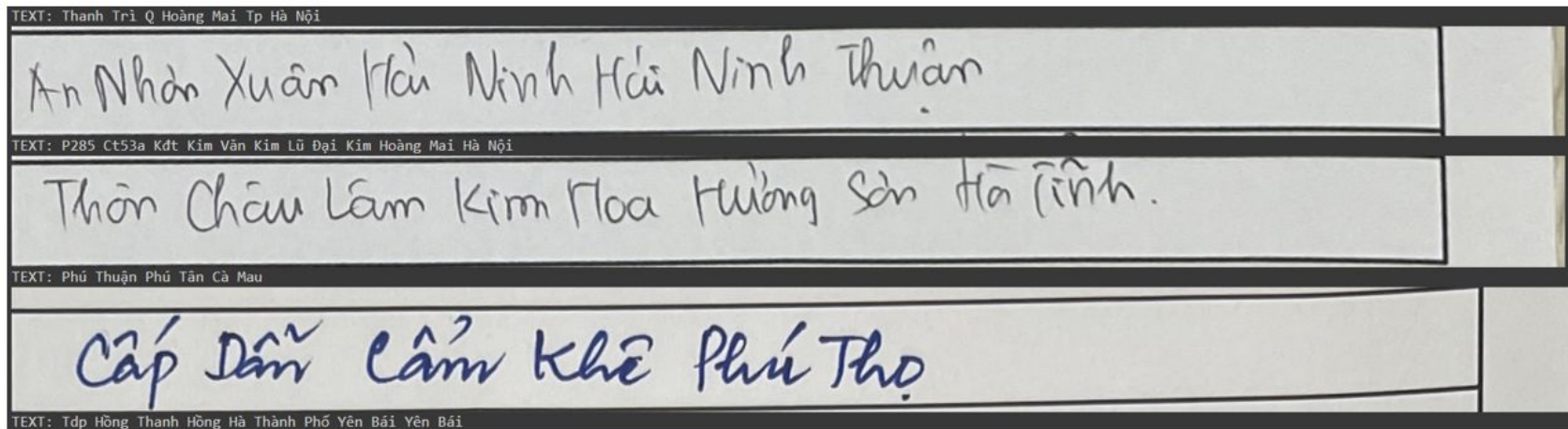
- 25/09/2023: **Công bố cuộc thi**
- 15/10/2023: **Bắt đầu vòng 1 của cuộc thi**
- 19/11/2023: **Nộp bài vòng 1**
- 23/11/2023: **Công bố kết quả vòng 1**
- 13/12/2023: **Nộp bài vòng 2**
- 16/12/2023: **Chung kết**

## Giải thưởng

- 02 giải Nhất, mỗi giải trị giá 20.000.000 VNĐ
- 02 giải Nhì, mỗi giải trị giá 10.000.000 VNĐ
- 04 giải Ba, mỗi giải trị giá 5.000.000 VNĐ

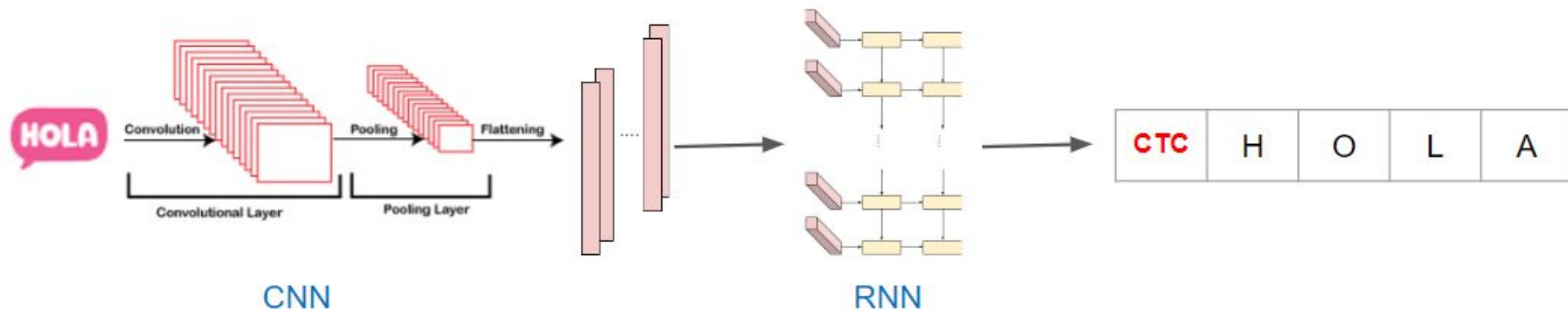
# Track2-Kalapa

Data - 2700 ảnh



# Track2-Kalapa

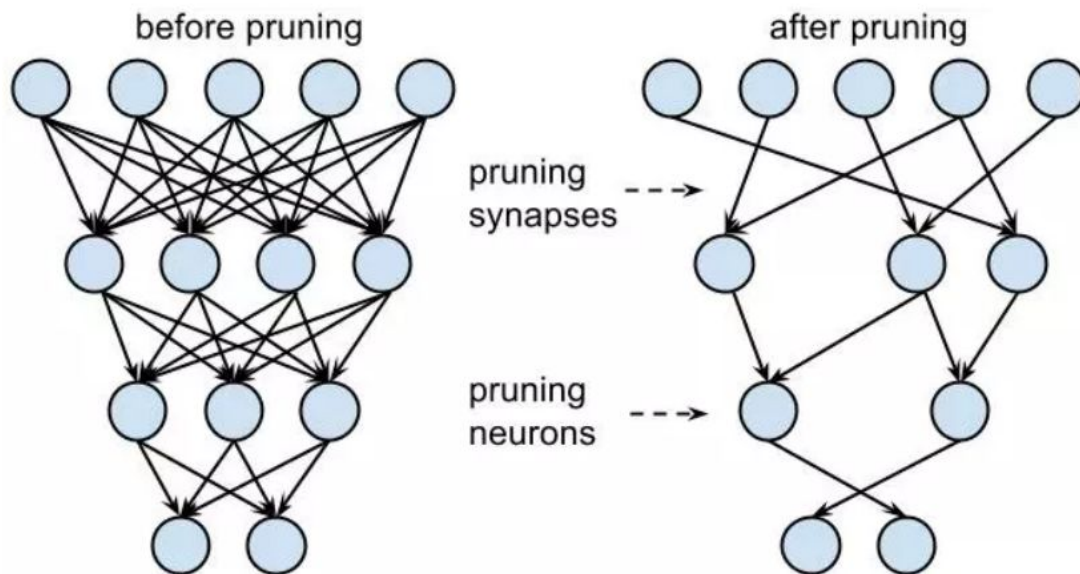
## Pipeline



Mô hình CRNN

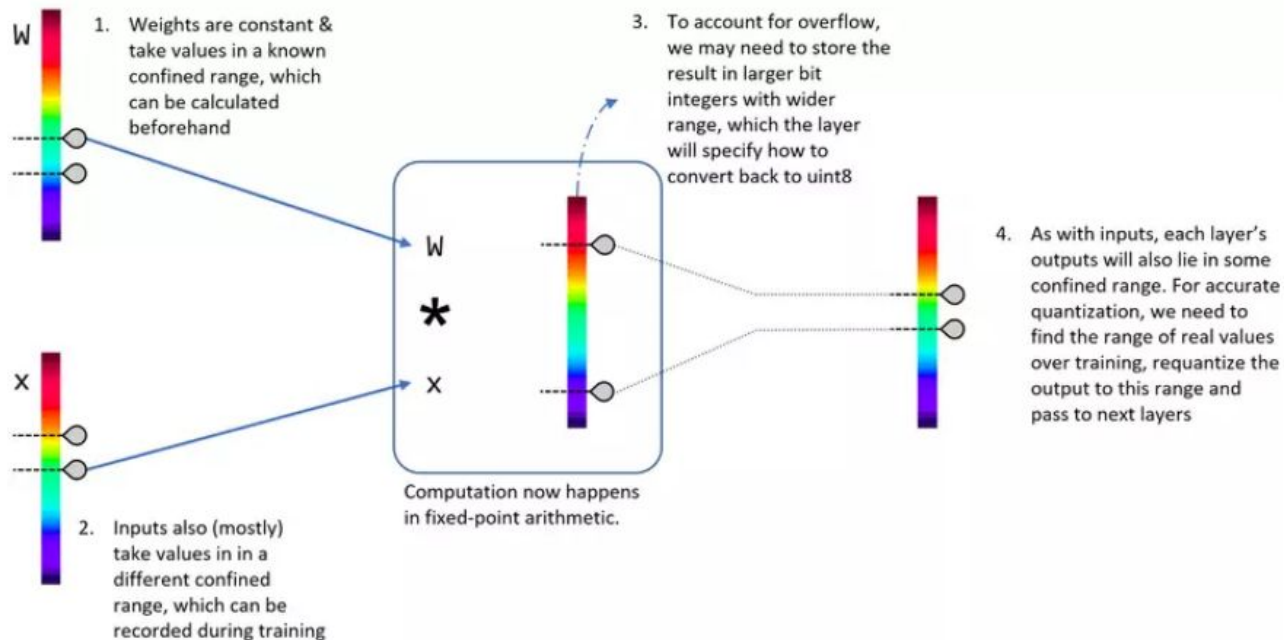
# Track2-Kalapa

## Model Compression - Network Pruning



# Track2-Kalapa

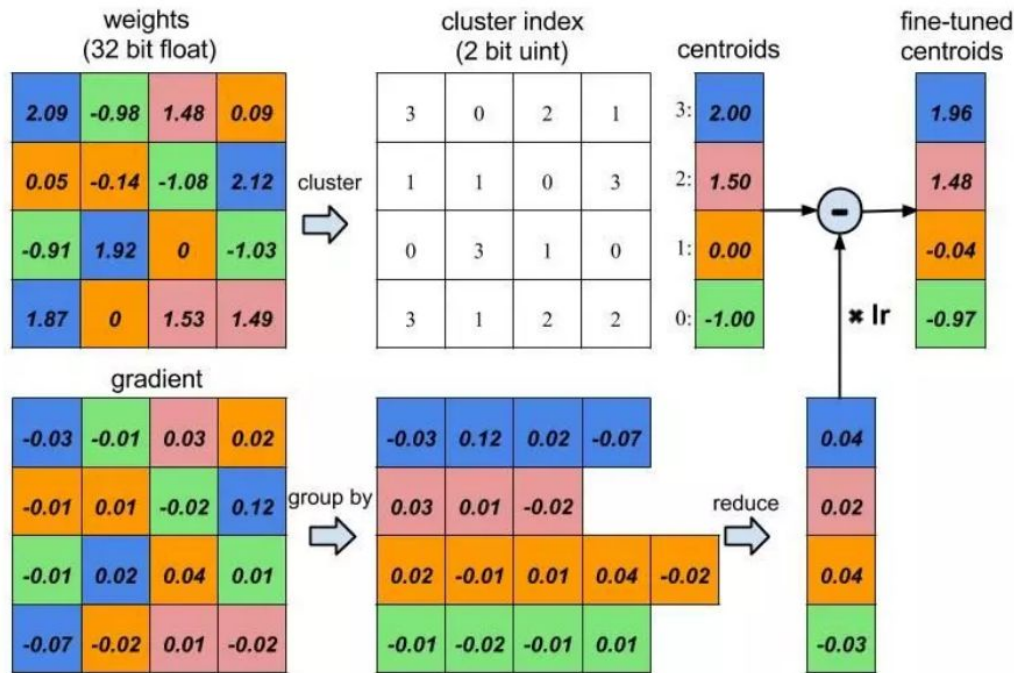
## Model Compression - Network Quantization





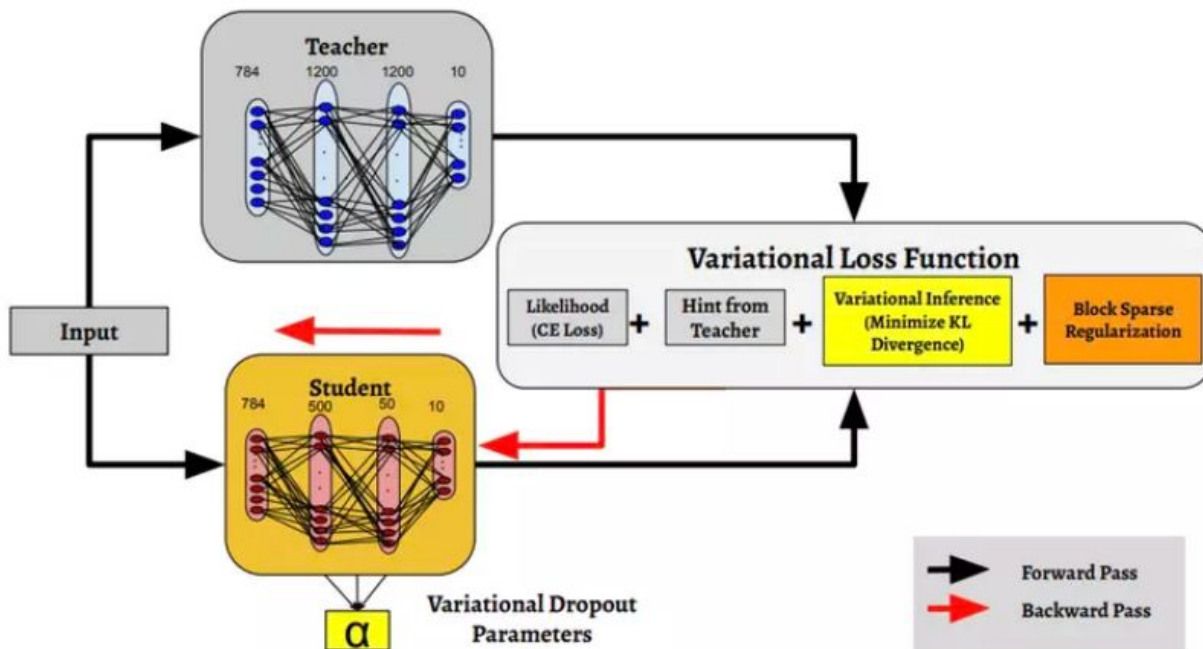
# Track2-Kalapa

## Model Compression - Weight-Sharing



# Track2-Kalapa

## Model Compression - Knowledge Distillation (KD)



# Track2-Kalapa

## Model Compression - PACT - [Paper](#)

Bitwidth	Original Paper ERR	Reproduced result ERR
5 bitwidth	8.4%	8.4%
4 bitwidth	9.2%	9.11%
3 bitwidth	9.4%	9.91%
2 bitwidth	11.1%	12.93%

PACT: Parameterized Clipping Activation for Quantized Neural Networks

The diagram illustrates the PACT-Quant framework pipeline, which consists of three main stages: Text Detection, Detection Boxes Rectify, and Text Recognition, each followed by a list of components. The final output is the detected text.

**Text Detection (db\_mv3\_slim, 1.4M)**

1. Light Backbone
2. Light Head
3. Remove SE
4. Cosine Learning Rate Decay
5. Learning Rate Warm-up
6. FPGM Pruner

**Detection Boxes Rectify (dir\_cls\_mv3\_slim, 0.5M)**

1. Light Backbone
2. Data Augmentation
3. Input Resolution
4. PACT Quantization

**Text Recognition (crnn\_mv3\_slim, 1.5M)**

1. Light Backbone
2. Data Augmentation
3. Cosine Learning Rate Decay
4. Feature Map Resolution
5. Regularization Parameters
6. Learning Rate Warm-up
7. Light Head
8. Pre-trained Model
9. PACT Quantization

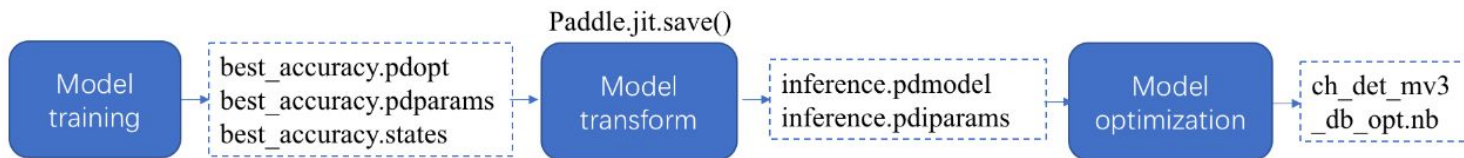
The final output is the detected text, shown as a list of text boxes on the right side of the diagram.

# Track2-Kalapa

## PPOCR - v2 - List model

model type	model format	description
inference model	inference.pdmodel, inference.pdiparams	Used for inference based on Paddle inference engine, <a href="#">detail</a>
trained model, pre-trained model	*.pdparams、 *.pdopt、 *.states	The checkpoints model saved in the training process, which stores the parameters of the model, mostly used for model evaluation and continuous training.
nb model	*.nb	Model optimized by Paddle-Lite, which is suitable for mobile-side deployment scenarios (Paddle-Lite is needed for nb model deployment).

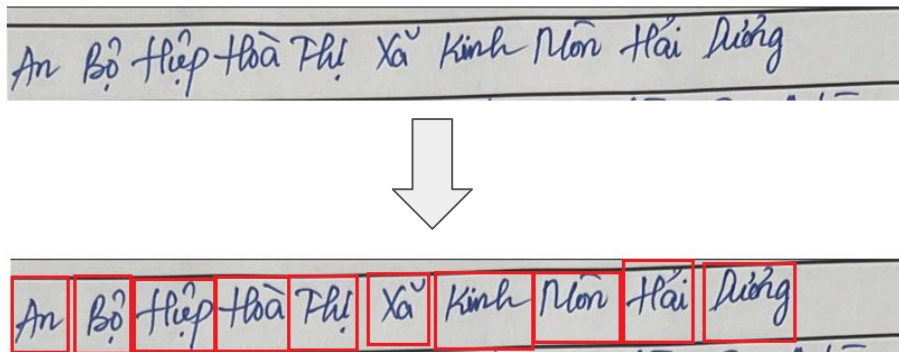
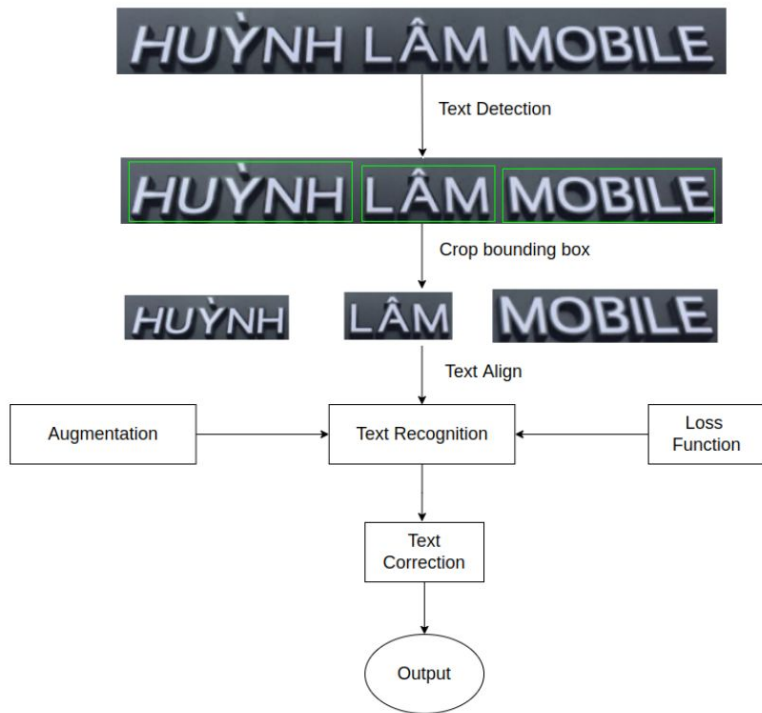
Relationship of the above models is as follows.



[Link List-model](#)

# Track2-Kalapa

## PPOCR - v2 - Pipeline Detect + Recognition



# Track2-Kalapa

Một số lưu ý:

- Không nên sử dụng ensemble
- Hậu xử lý bằng cách tìm từ tương đồng (trong vocab)
- Hậu xử lý bằng cách xây dựng tập luật Tiếng Việt
- Sử dụng augmentation

# Track2-Kalapa

## Fuzzy Search - [Link](#)

### Partial Ratio [↗](#)

```
>>> fuzz.partial_ratio("this is a test", "this is a test!")  
100
```



### Token Sort Ratio [↗](#)

```
>>> fuzz.ratio("fuzzy wuzzy was a bear", "wuzzy fuzzy was a bear")  
91  
>>> fuzz.token_sort_ratio("fuzzy wuzzy was a bear", "wuzzy fuzzy was a bear")  
100
```

