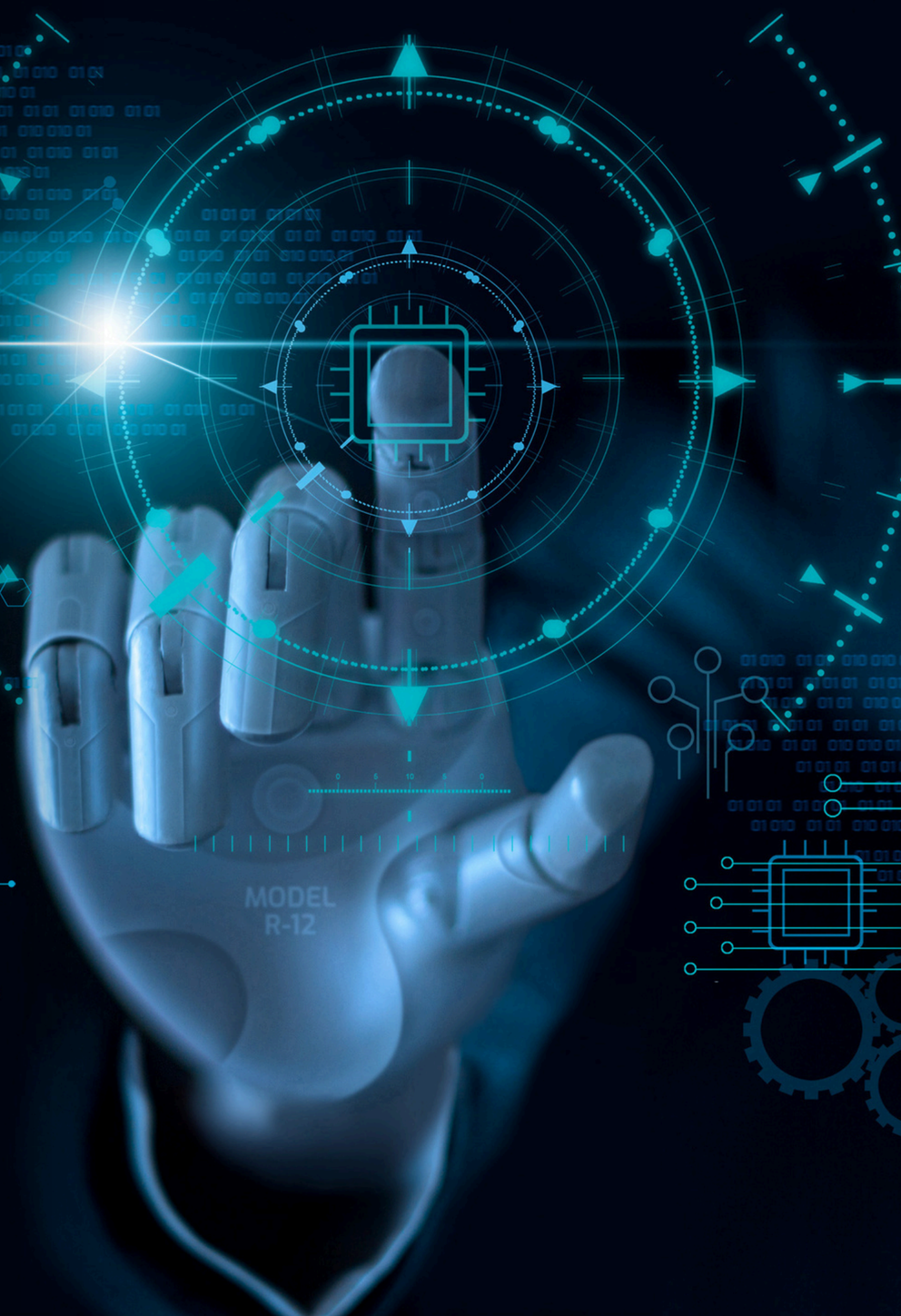
ESIEE PARIS

P2PNet for Crowd Counting

A Point-to-Point
Regression Framework

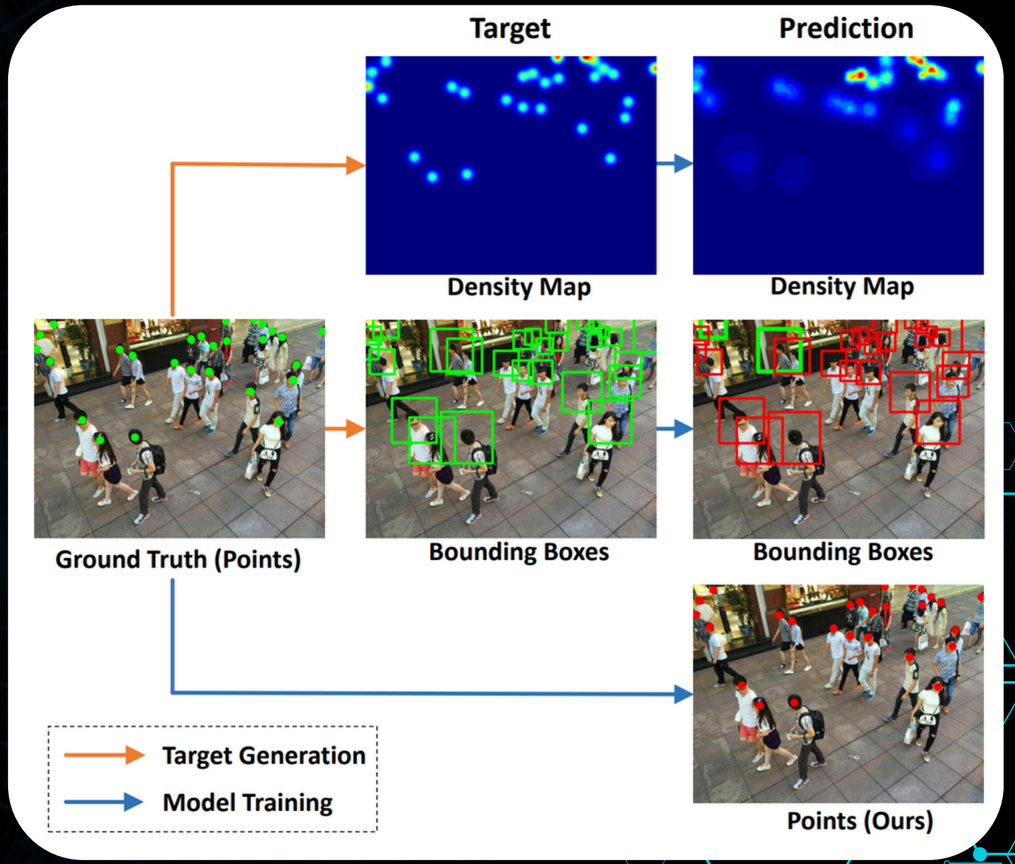
Presented By **PENG Zhenzhen & Andrei STUPU**

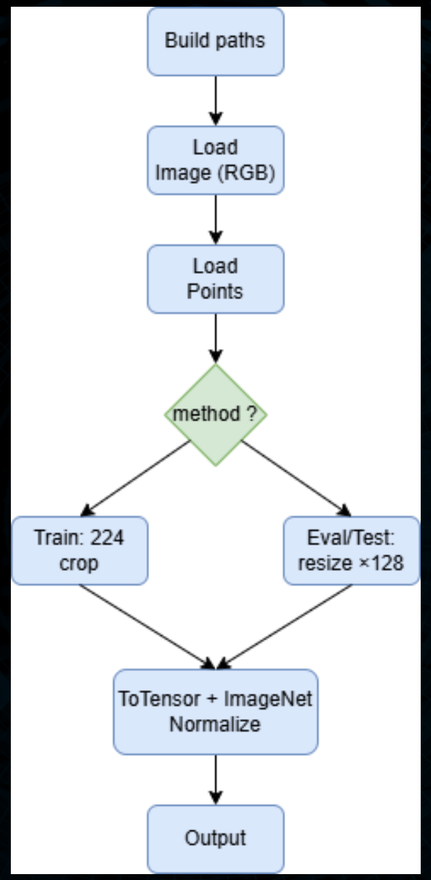


Project Objective & Background

Counting people in dense crowds is hard (Occlusion, Scale variation)

Traditional methods estimate density maps, which can be blurry. Our objective is to implement P2PNet, which directly predicts the exact (x, y) coordinate for each person. This is an end-to-end approach that simplifies the pipeline.





Dataset & Preprocessing

Creativity Unleashed

Dataset: ShanghaiTech Part A (Highly congested scenes).

Source: Internet images, variable resolutions.

Preprocessing (dataset.py):

Fixed Size Training: Random Crop to 224x224 and (Data Augmentation).

Augmentation: added random flip and jitter

Testing: Resize to multiples of 128 (to fit VGG16 strides), but keep original aspect ratio.

Neural Network Architecture

Revolutionizing Health and Wellness

- **Backbone:** VGG16-BatchNorm (backbone.py). Uses features from different layers (C3, C4, C5).
- **Neck:** FPN (Feature Pyramid Network) to fuse deep (semantic) and shallow (detail) features.
- **Heads:** Two branches for every point:

1. Regression: Where is the point? (▵x, ▵y)
2. 2. Classification: Is this a person? (Score)

- Our model architecture, defined in p2pnet.py, uses a VGG16 backbone. We extract features at strides 8 and 16. These features are fed into two parallel heads: one predicts the coordinate offset, and the other predicts the confidence score.

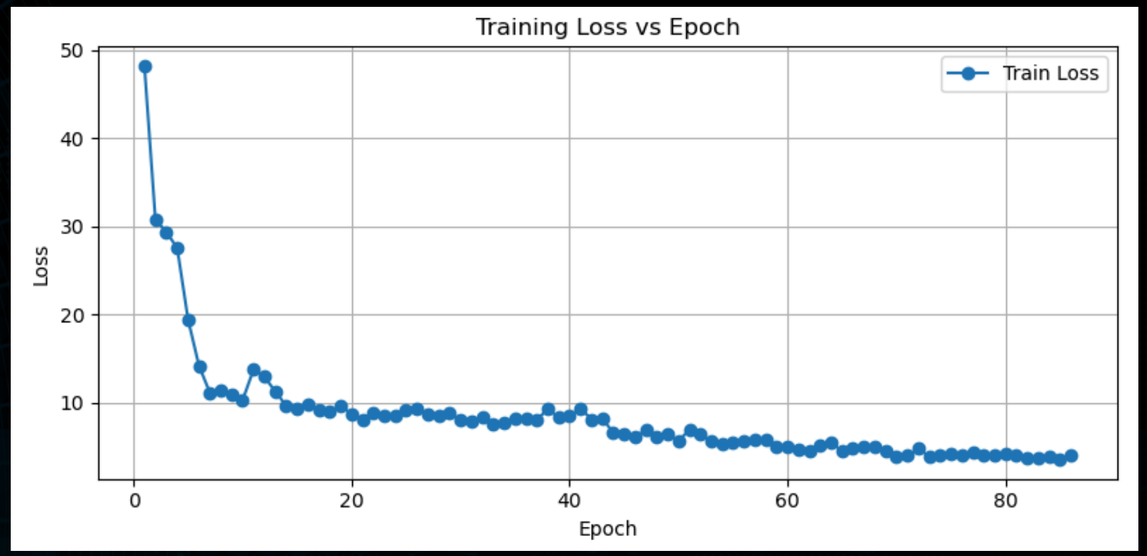
The "Hungarian" Matcher

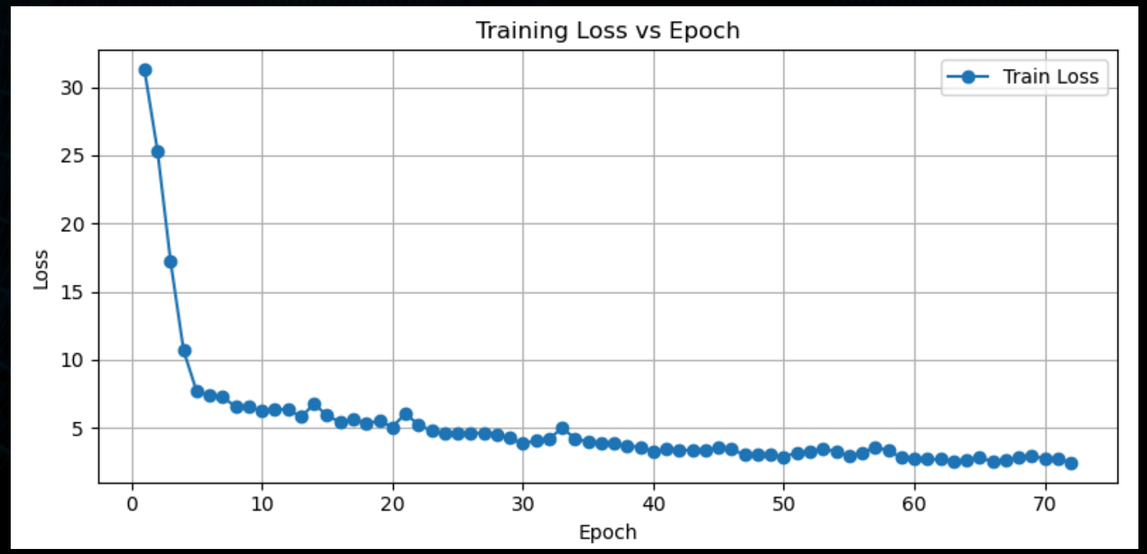
The Rise of Smart Machines

- The Problem: Model predicts 10,000 points, but there are only 200 people. Which prediction corresponds to which person?
- The Solution: One-to-One Matching (matcher.py).
- Mechanism: Hungarian Algorithm minimizes the global cost (Classification Score + Point Distance).

Loss Function & Training

- Lcls: BCE with logits (Focus on matched points).
- Lpoint: MSE Loss (Only for matched points).



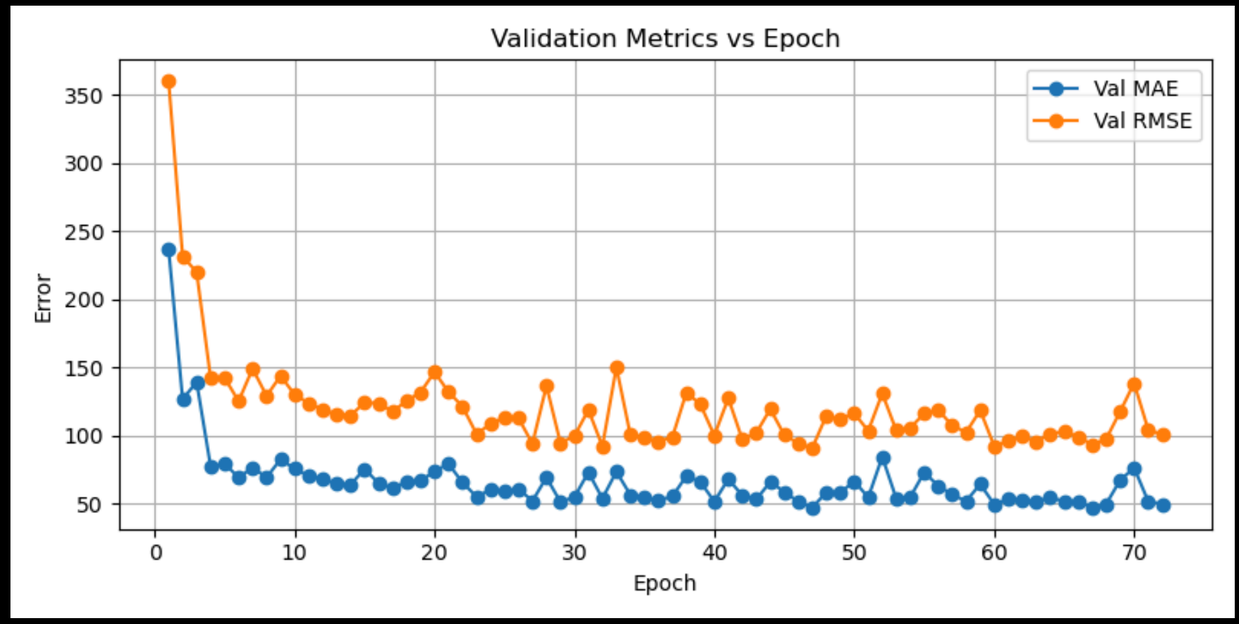


Fine Tuning

Freezed the backbone first to stabilize the training.

unfreeze later with a smaller LR (backbone).

Early stopping on validation MAE to prevent overfitting.



Evaluation & Conclusion

There are some “good” results…

…and some problems

Conclusion & Future Work

- Summary: P2PNet is effective, end-to-end, and removes complex post-parocessing.
- Future: Try stronger backbones (ResNet-50 or Swin Transformer). Improving detection on extremely small scales.
- Add localization metrics (Precision/Recall)

Thank You !