Information Technologies for Industrial Engineers

เทคโนโลยีสารสนเทศสำหรับวิศวกรอุตสาหการ

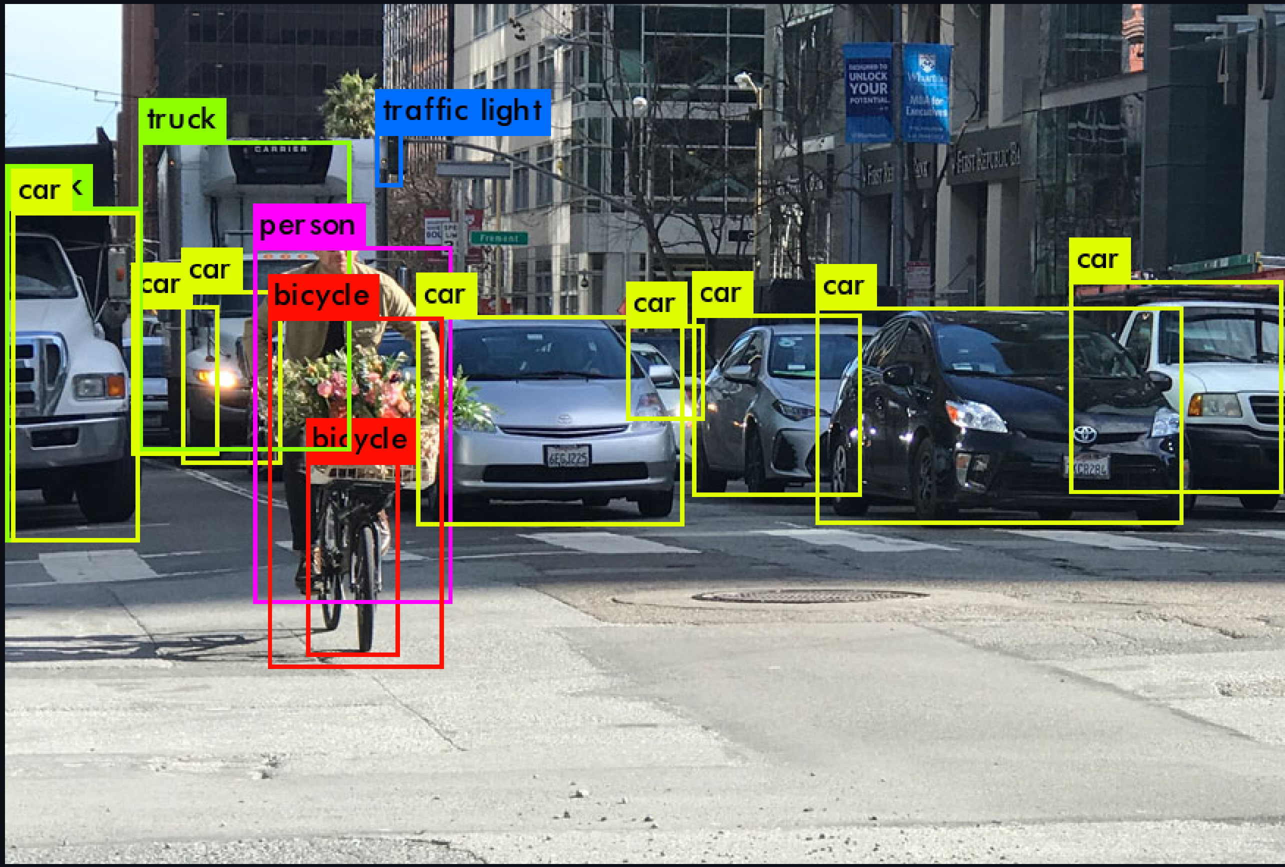
Object Detection Application

Image classification

- Dog (50%)
- … (15%)
- … (15%)

Object detection

- Car
  - Top: 500, Bottom: 200, Left: 50, right: 400
  - 50%
- Bicycle
  - …
  - …



Models

- **YOLO** (You Only Look Once)
  - Grid-based approach
  - Faster, less accurate
- **SSD** (Single Shot Detector)
  - Feature-map approach
  - Faster, less accurate (comparable to YOLO)
- **R-CNN** (Region-based Convolutional Neural Network)
  - Pixel classification
  - Slower, more accurate

Source

COCO dataset

- *Common Objects in Context*
- Large-scale image recognition dataset for object detection, segmentation, and captioning tasks.
    - Contains over 330,000 images.
    - Annotated with 80 object categories.
- https://cocodataset.org/#explore

COCO SSD

- This model detects objects defined in the COCO dataset.
- Uses SSD algorithm
- https://github.com/tensorflow/tfjs-models/tree/master/coco-ssd

Object detection app (image file)

Setting up

- `npm create vite@latest`
- ...
- `npm install @tensorflow/tfjs @tensorflow-models/coco-ssd`

Code

- `./src/model.ts` 🔗
- `./src/app.tsx` 🔗

Object detection app (webcam)

Setting up

- `npm create vite@latest`
- ...
- `npm install @tensorflow/tfjs @tensorflow-models/coco-ssd react-webcam`
  - Note the additional `react-webcam` package.

Utilitiy files

- `./src/model.ts` 🔗
- `./src/App.css` 🔗
- `./src/utils.ts` 🔗

Main program

`./src/App.tsx` 🔗

Webcam + Prediction Display

- Canvas overlays the webcam.

./src/App.tsx

```
<div className="wrapper"> //position: relative
  <canvas id="canvas" className="canvas" /> //  position: absolute;
  <div className="webcam"> //  position: absolute;
    <Webcam ... />
  </div>
</div>
```

Single prediction

`./src/App.tsx`

```
async function singlePrediction() {
  const predictions = await getPrediction(model, webcamRef);
  setPredictions(predictions);
  if (predictions) displayPredictions(predictions, width, height);
}
```

displayPrediction

`./src/utils.tsx`

```
export function displayPredictions(predictions, width, height) {
  var canvas = document.getElementById("canvas");
  const ctx = canvas.getContext("2d");
  // ...
  predictions.forEach((prediction) => {
    drawBox(prediction, ctx);
  });
}
```

drawBox

./src/utils.ts

```
function drawBox(prediction, ctx) {
  let bboxLeft = prediction.bbox[0];
  let bboxTop = prediction.bbox[1];
  let bboxWidth = prediction.bbox[2];
  let bboxHeight = prediction.bbox[3]; // - bboxTop;

  // ...
  ctx.rect(bboxLeft, bboxTop, bboxWidth, bboxHeight);
  // ...
}
```

Continuous prediction

./src/App.tsx

```tsx
const sub = useRef<any>(null);

function continuousPrediction() {
  setIsPredicting(true);
  sub.current = setInterval(async () => {
    const predictions = await getPrediction(model, webcamRef);
    setPredictions(predictions);
    if (predictions) displayPredictions(predictions, width, height);
  }, 1000);
}

function stopPrediction() {
  setIsPredicting(false);
  if (sub.current) clearInterval(sub.current);
}
```

Development with mobile

- Visit https://ngrok.com
  - Sign up
  - Sign in
  - Verify email
- Go to `Cloud Edge` -> `Domains`
  - Create a new domain
  - Let's call it `DOMAIN_NAME`.
- Download software and extract to `Desktop`

- Open Command Prompt and nagivate to `Desktop`
  - `cd Desktop`
- Authenticate
  - `ngrok config add-authtoken AUTH_TOKEN`
  - *`AUTH_TOKEN` is from the website.*
- Run
  - `ngrok http --domain=DOMAIN_NAME 5173`
- Open your `DOMAIN_NAME` in a mobile browser.
  - Do not use Line browser.

> In PowerShell, you need to use `./ngrok ...`