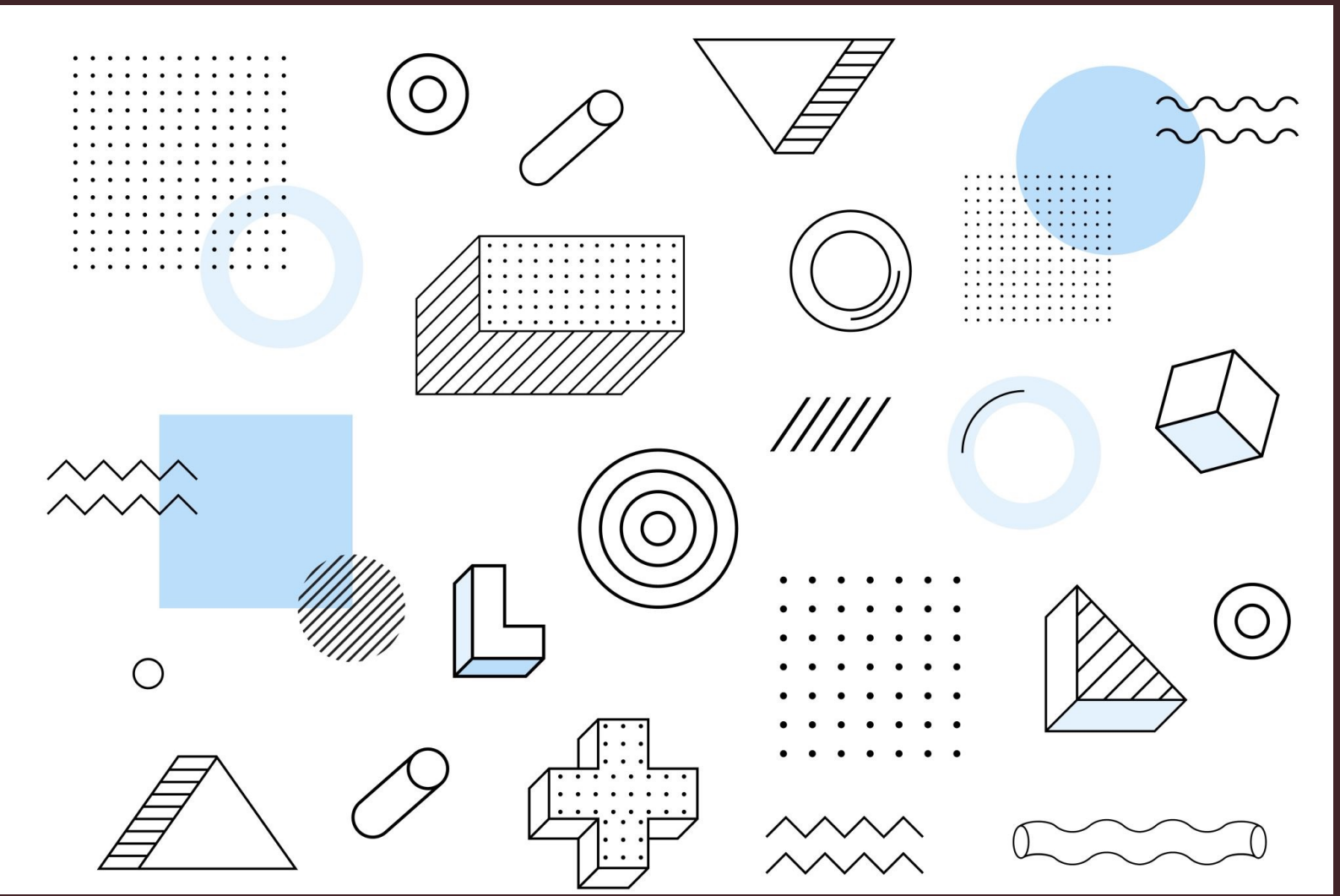


NUMBER PLATE RECOGNITION

Hồ Thiên Phước

Lưu Thị Hồng Ngọc

ĐỘNG LỰC NGHIÊN CỨU

PHÁT BIỂU BÀI TOÁN

BẢNG HIỆU NĂNG

NGUYÊN LÍ-PHƯƠNG PHÁP-GIẢI THUẬT

CÁCH CÀI ĐẶT

ĐỘNG LỰC NGHIÊN CỨU- Ý NGHĨA KHOA HỌC

-TÌM HIỂU PHƯƠNG PHÁP PHÁT HIỆN BIÊN CẠNH

-GÁN NHÃN VỚI LABELIMG TOOL (CHO TRƯỚC TOẠ ĐỘ ĐIỂM Ở 4 GÓC KHUNG BIỂN SỐ)

..

ĐỘNG LỰC NGHIÊN CỨU-ỨNG DỤNG

-ĐỀ TÀI ĐƯỢC GIAO LÀ 1 BÀI TOÁN ỨNG DỤNG THỰC TẾ QUEN THUỘC ĐƯỢC ỨNG DỤNG RỘNG RÃI.

- Nhu cầu để giám sát an ninh, bãi đỗ xe thông minh, hệ thống thu tiền không dừng tại các tuyến đường cao tốc…

PHÁT BIỂU BÀI TOÁN-

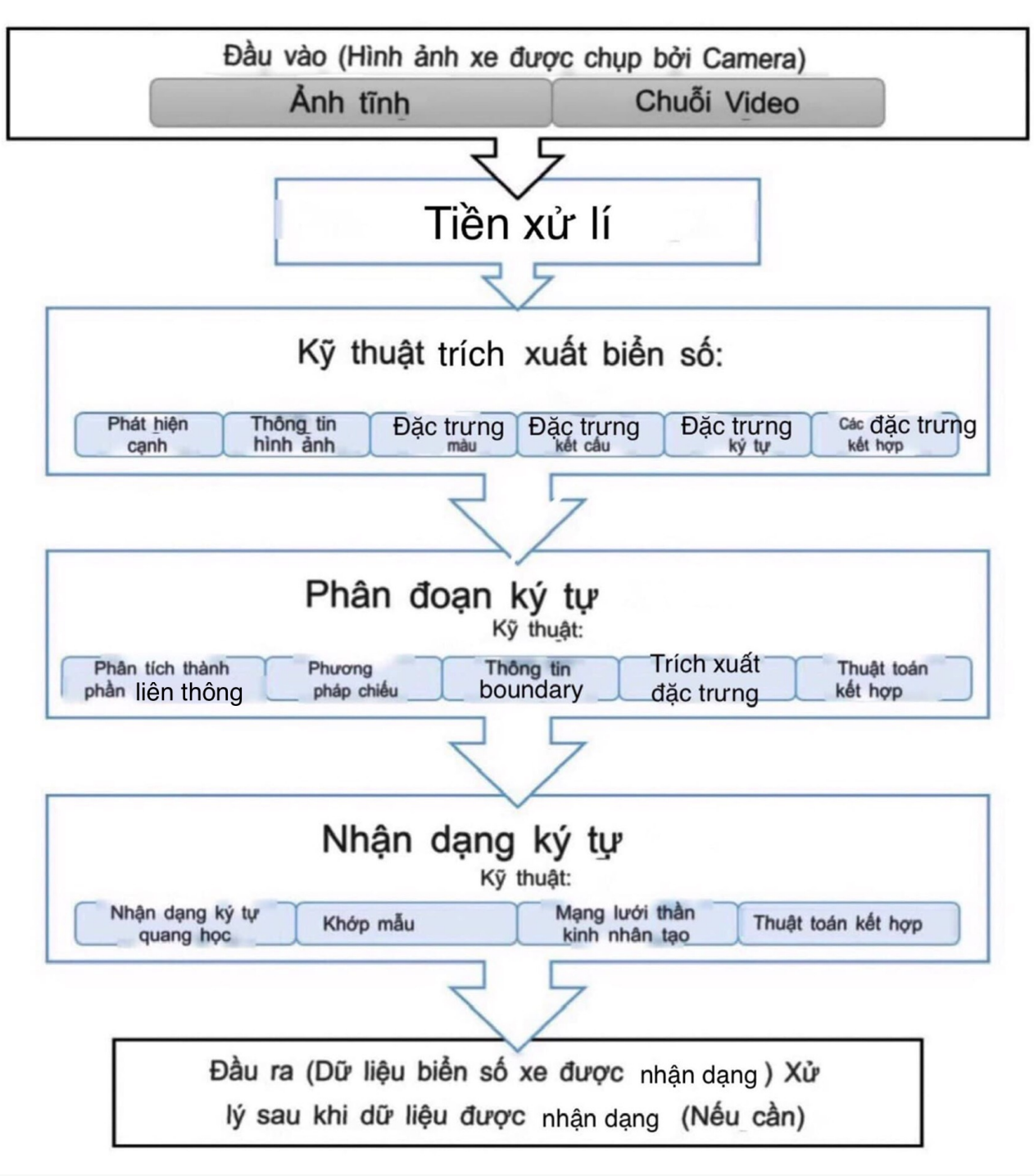
INPUT

OUTPUT

Input: Hình ảnh/video có chứa biển số.

Output: Vị trí bounding box của khung biển số và class id là thông tin chữ và số có trong khung biển số ở input.

PHÁT BIỂU BÀI TOÁN- FRAMEWORK



PHÁT BIỂU BÀI TOÁN- THÁCH THỨC

QUÁ TRÌNH TRAINING TRẢI QUA NHIỀU CÔNG ĐOẠN VÀ THỰC THI KÉO DÀI SUỐT NHIỀU TIẾNG ĐỒNG HỒ

BẢNG HIỆU NĂNG

XEM THÊM TẠI MỤC CÔNG TRÌNH LIÊN QUAN TRONG REPORT ĐÍNH KÈM

NGUYÊN LÍ

GÁN NHÃN VỚI LABELIMG (CHO TRƯỚC TOẠ ĐỘ ĐIỂM 4 GÓC BIỂN SỐ) ĐỂ TRÍCH XUẤT KHUNG

BOUNDARY BAO TRỌN ĐỐI TRƯỢNG ĐỂ PHÂN ĐOẠN KÍ TỰ

NHẬN DẠNG= PHÁT HIỆN VÀ PHÂN LỚP

PHƯƠNG PHÁP

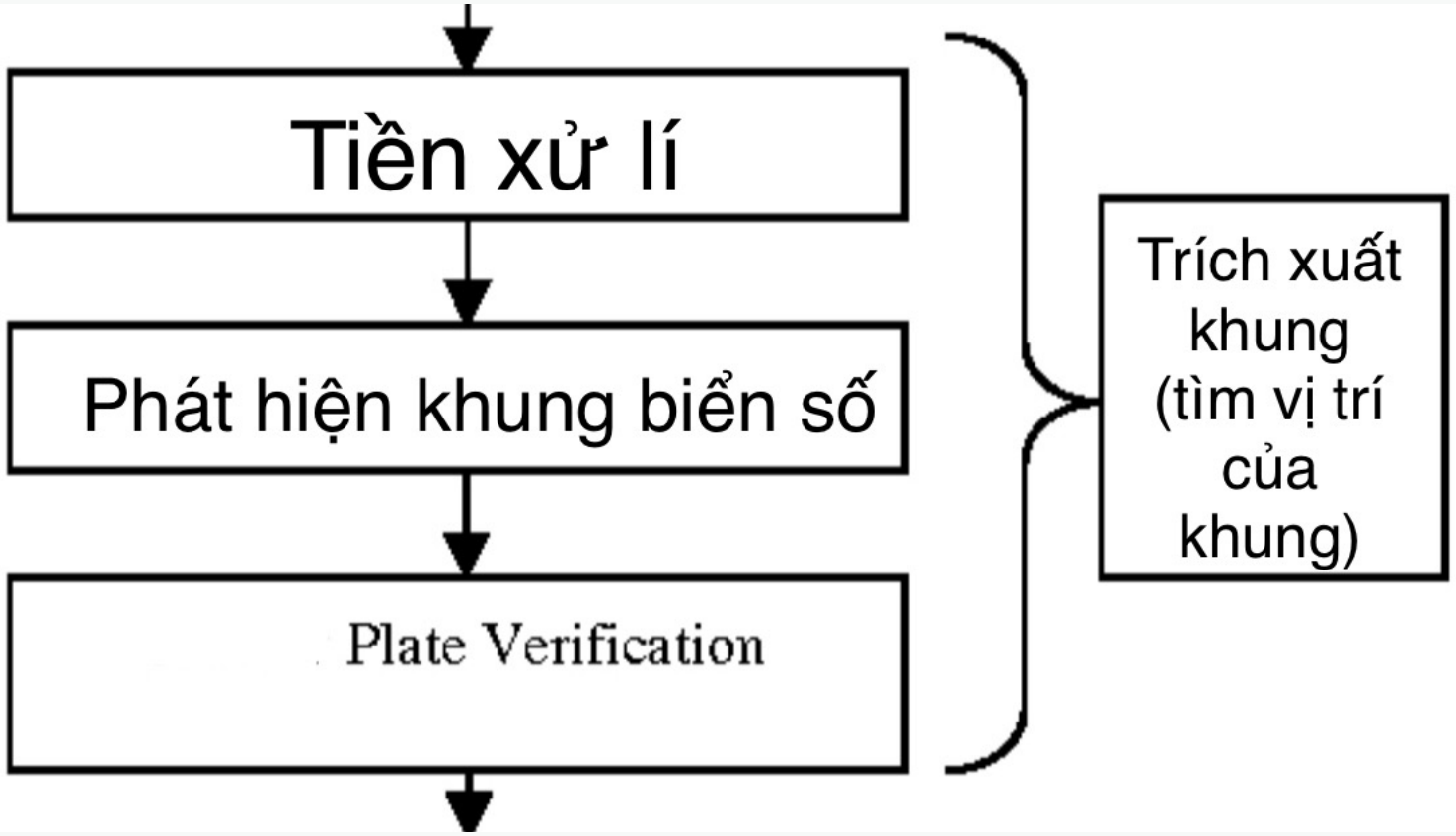
B1: Xác định vùng chứa biển số xe sử dụng Yolo Tiny v3

B2: Sử dụng thuật toán segment để tách từng kí tự trên biển số xe

B3: Xây dựng một model CNN để phân loại các kí tự (characters classification)

B4: Định dạng lại biển số xe xác định biển số xe gồm một hay hai dòng.

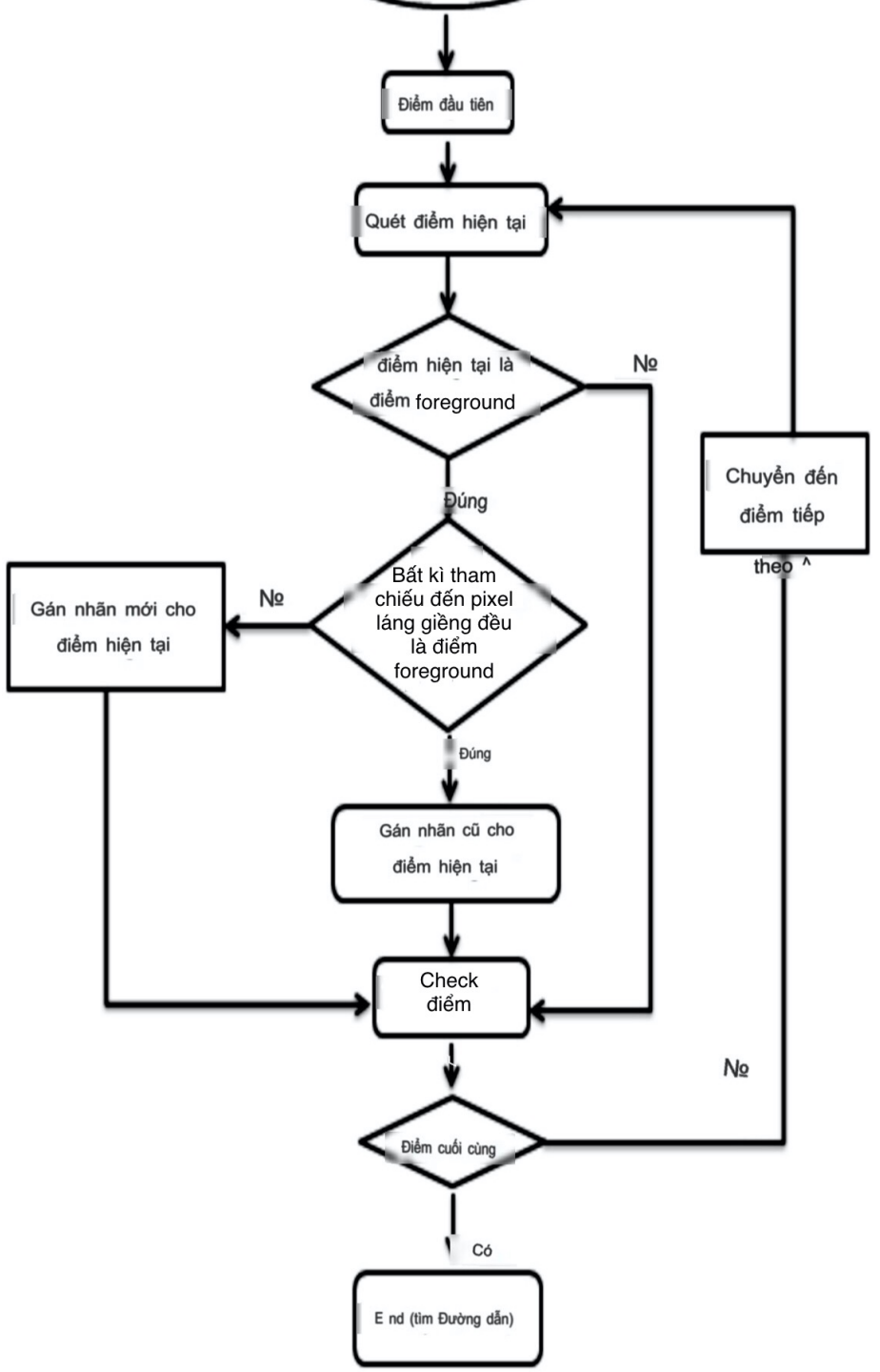
PHƯƠNG PHÁP-TRÍCH XUẤT KHUNG BIỂN SỐ

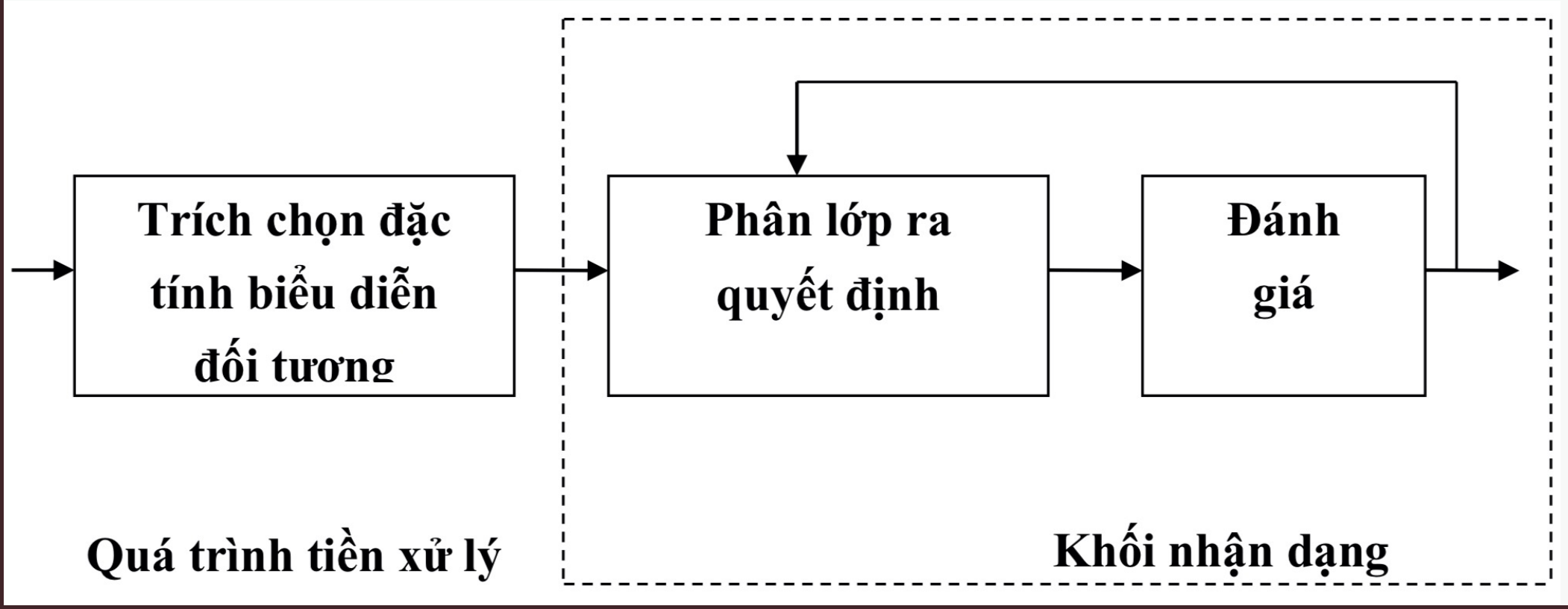


PHƯƠNG PHÁP-PHÂN ĐOẠN KÍ TỰ

Connected components analysis (Phân tích thành phần liên thông)

Thuật toán này có ý tưởng đơn giản là nó sẽ kết nối tất cả pixel nào có cùng giá trị thành một khối và gắn cho nó một cái nhãn(label). Nhờ đó tất cả các pixel của cùng một kí tự và có cùng giá trị sẽ được kết nối và được tách ra khỏi biển số xe.





PHƯƠNG PHÁP- NHẬN DẠNG KÍ TỰ

GIẢI THUẬT

GIAI ĐOẠN 1: TRAINING

(LINK HƯỚNG DẪN , LINK HƯỚNG DẪN 2)

GIAI ĐOẠN 2: DÙNG .WEIGHTS THU ĐƯỢC SAU KHI TRAIN ĐỂ NHẬN DẠNG

GIẢI THUẬT-

TRAINING (CHO YOLOV4)

XEM Ở REPORT
VÀ .IPYNB ĐÍNH
KÈM

GIẢI THUẬT-

NHẬN DẠNG

Xác định vùng chứa biển số xe sử dụng Yolo Tiny v3

```
from imutils import perspective
# crop number plate used by bird's eyes view transformation
LpRegion = perspective.four_point_transform(image_original, pts)
```

GIẢI THUẬT-

NHẬN DẠNG

Segment tách từng kí tự trên biến số xe

```
V = cv2.split(cv2.cvtColor(LpRegion, cv2.COLOR_BGR2HSV))[2]
```

```
from skimage.filters import threshold_local
```

```
T = threshold_local(V, 15, offset=10, method="gaussian")
```

```
thresh = (V > T).astype("uint8") * 255
```

GIẢI THUẬT-

NHẬN DẠNG

Segment tách từng kí tự trên biển số xe

```
# convert black pixel of digits to white pixel
thresh = cv2.bitwise_not(thresh)
cv2.imwrite("step2_2.png", thresh)
thresh = imutils.resize(thresh, width=400)
thresh = cv2.medianBlur(thresh, 5)
# connected components analysis
labels = measure.label(thresh, connectivity=2, background=0)
```

```
# loop over the unique components
for label in np.unique(labels):
  # if this is background label, ignore it
  if label == 0:
    continue
```

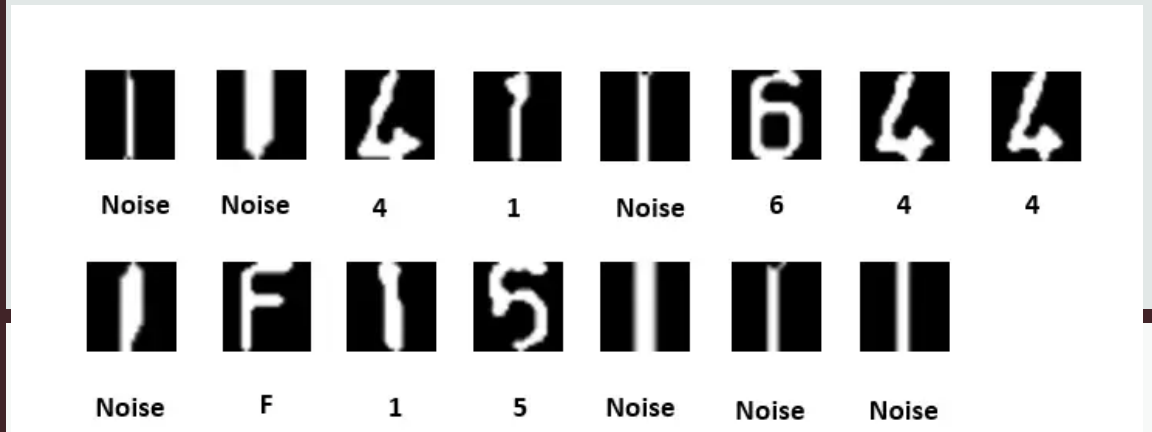
```
  # init mask to store the location of the character candidates
  mask = np.zeros(thresh.shape, dtype="uint8")
  mask[labels == label] = 255

  # find contours from mask
  contours, hierarchy = cv2.findContours(mask,
cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)

  if len(contours) > 0:
    contour = max(contours, key=cv2.contourArea)
    (x, y, w, h) = cv2.boundingRect(contour)

    # rule to determine characters
    aspectRatio = w / float(h)
    solidity = cv2.contourArea(contour) / float(w * h)
    heightRatio = h / float(LpRegion.shape[0])

    if 0.1 < aspectRatio < 1.0 and solidity > 0.1 and 0.35 <
heightRatio < 2.0:
        # extract characters
        candidate = np.array(mask[y:y + h, x:x + w])
        square_candidate = convert2Square(candidate)
        square_candidate = cv2.resize(square_candidate, (28,
28), cv2.INTER_AREA)
        square_candidate = square_candidate.reshape((28, 28,
1))
        self.candidates.append((square_candidate, (y, x)))
```



GIẢI THUẬT-

NHẬN DẠNG

Phân loại các kí tự sử dụng model CNN

```
ALPHA_DICT = {0: 'A', 1: 'B', 2: 'C', 3: 'D', 4: 'E', 5: 'F', 6: 'G', 7: 'H', 8: 'K', 9: 'L', 10: 'M', 11: 'N', 12: 'P',
              13: 'R', 14: 'S', 15: 'T', 16: 'U', 17: 'V', 18: 'X', 19: 'Y', 20: 'Z', 21: '0', 22: '1', 23: '2', 24: '3',
              25: '4', 26: '5', 27: '6', 28: '7', 29: '8', 30: '9', 31: "Background"}
```

```
def _build_model(self):
  # CNN model
  self.model = Sequential()
  self.model.add(Conv2D(32, (3, 3), padding='same', activation='relu', input_shape=(28, 28, 1)))
  self.model.add(Conv2D(32, (3, 3), activation='relu'))
  self.model.add(MaxPooling2D(pool_size=(2, 2)))
  self.model.add(Dropout(0.25))

  self.model.add(Conv2D(64, (3, 3), padding='same', activation='relu'))
  self.model.add(Conv2D(64, (3, 3), activation='relu'))
  self.model.add(MaxPooling2D(pool_size=(2, 2)))
  self.model.add(Dropout(0.25))

  self.model.add(Conv2D(64, (3, 3), padding='same', activation='relu'))
  self.model.add(Conv2D(64, (3, 3), activation='relu'))
  self.model.add(MaxPooling2D(pool_size=(2, 2)))
  self.model.add(Dropout(0.25))

  self.model.add(Flatten())
  self.model.add(Dense(512, activation='relu'))
  self.model.add(Dropout(0.5))
  self.model.add(Dense(32, activation='softmax'))
```

GIẢI THUẬT

XEM ĐẦY ĐỦ Ở SOURCE CODE ĐÍNH KÈM

CÁCH CÀI ĐẶT- MÔI TRƯỜNG

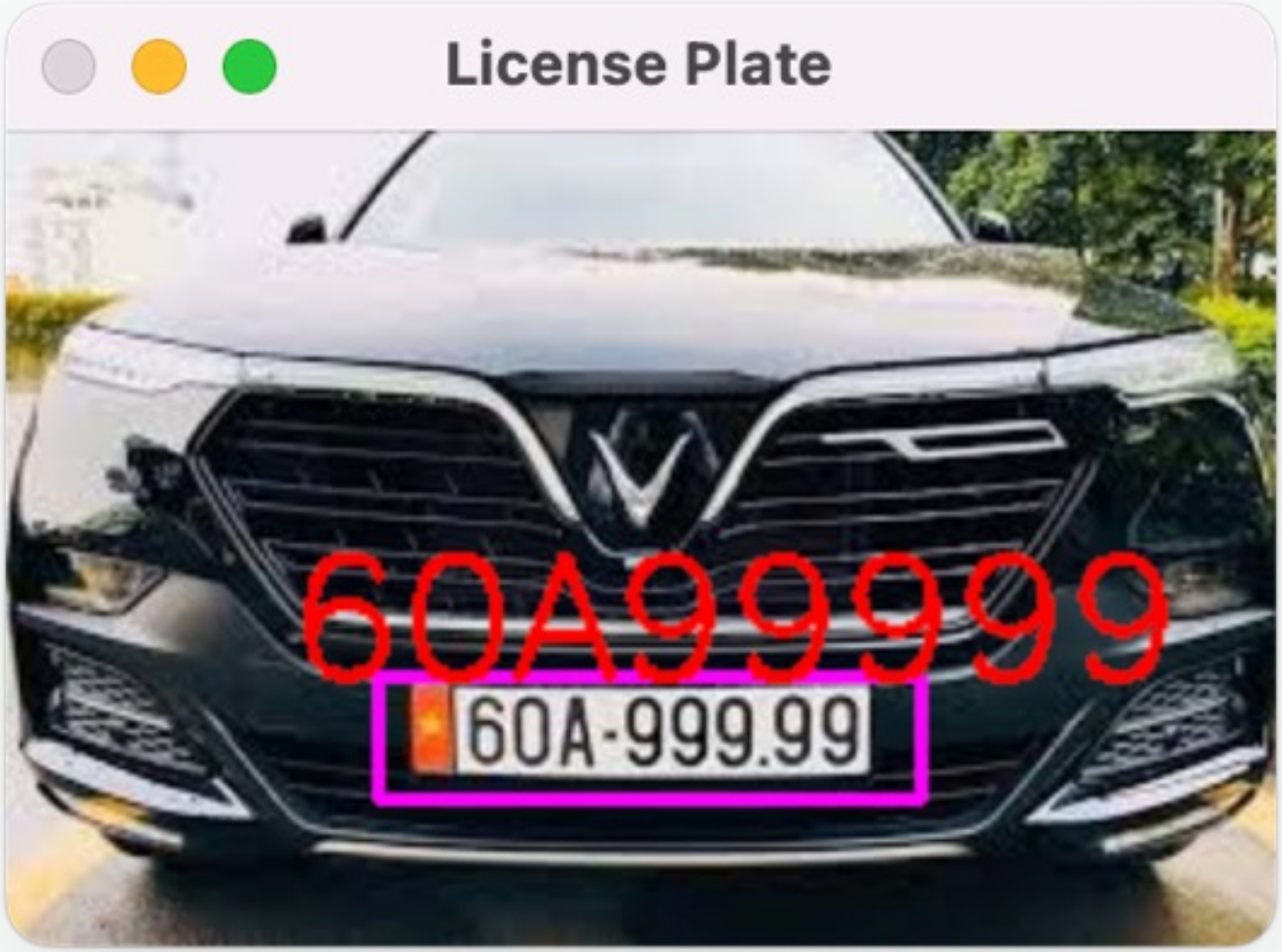
CÔNG CỤ: GOOGLE COLAB, VISUAL STUDIO CODE

NGÔN NGỮ: PYTHON

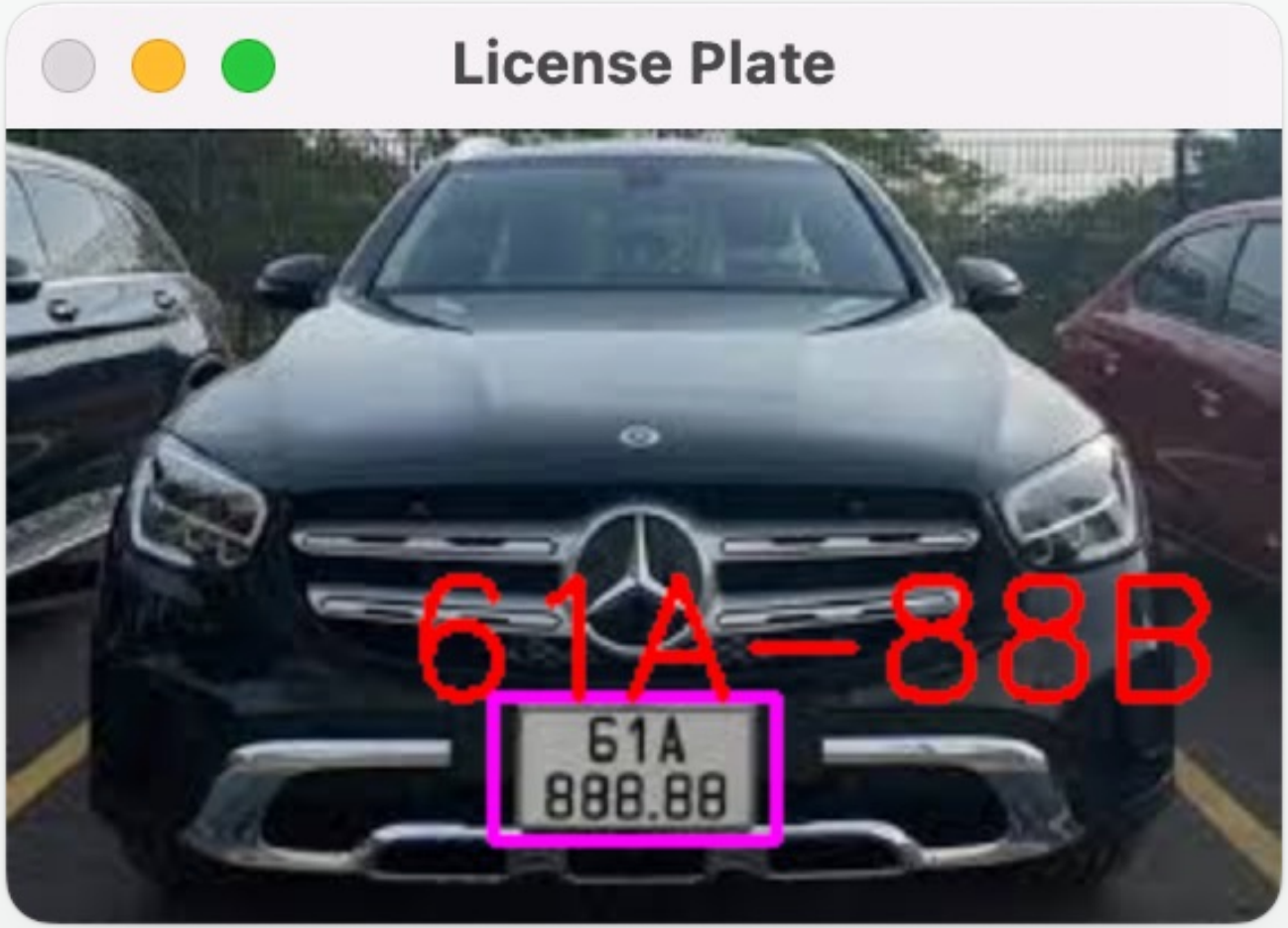
CÁCH CÀI ĐẶT- MÔI TRƯỜNG PYTHON VENV VỚI VSCODE

https://courses.ctda.hcmus.edu.vn/mod/resource/view.php?id=51436

CÁCH CÀI ĐẶT- KẾT QUẢ



CÁCH CÀI ĐẶT- KẾT QUẢ



XIN CẢM ƠN