



```
class CustomPEKModel(MagicPDF):  
    def __init__(self, min_box: float, show_log: bool = False, **kwargs):  
        super().__init__(min_box, show_log, **kwargs)  
        self.min_box = min_box  
        self.show_log = show_log
```

```
class ModifiedPaddleOCR(PaddleOCR):  
    def __init__(self, *args, **kwargs):  
        super().__init__(*args, **kwargs)  
        self.vietocr_cfg = Cfg.load_config_from_name('vgg_transformer')  
        self.vietocr_cfg['device'] = 'cpu'  # Force CPU device  
        self.vietocr_detector = Predictor(self.vietocr_cfg)  
  
    class RollFromPaddleOCR(PaddleOCR):  
        def ocr(self, img, det=True, rec=True, cls=True, bin=False, bin=False, mfd_res=None, **kwargs):  
            if det and rec:  
                ocr_res = []  
                for idx, img in enumerate(imgs):  
                    img = preprocess(img)  # self._preprocess(img, cls, mfd_res=mfd_res)  
                    if not det_boxes and not ocr_res.append(None)  
                    continue  
                    tmp_res = []  
                    for box, res in zip(det_boxes, rec_res):  
                        x_min = int(box[0][0])  
                        y_min = int(box[0][1])  
                        x_max = int(box[2][0])  
                        y_max = int(box[2][1])  
                        text_img = img[y_min:y_max, x_min:x_max]  
                        text = self.vietocr_detector.predict(image.fromarray(text_img))  
                        tmp_res.append((box.toList(), (text, res[1])))  
                    ocr_res.append(tmp_res)  
            return ocr_res
```

Modify hàm này với text recognition là vietOCR pretrained

trả về tọa độ các box sau khi xử lý và recog của box (không cần qam tới recog này vì recog này sử dụng paddleocr, ta chỉ cần tọa độ bbox, sau khi có tọa độ tiến hành recog lại = vietocr)

các hàm này nằm trong thư viện paddleOCR

```
def get_rotate_crop_image(img, points):  
    img_height, img_width = img.shape[0:2]  
    left = int(min(points[:, 0]))  
    right = int(max(points[:, 0]))  
    top = int(min(points[:, 1]))  
    bottom = int(max(points[:, 1]))  
    img_crop = img[top:bottom, left:right, :].copy()  
    points[:, 0] = points[:, 0] - left  
    points[:, 1] = points[:, 1] - top  
    points = expand_bounding_box(np.array(points))  
    assert len(points) == 4, "Shape of points must be 4x2"  
    points = np.float32(points)  
    img_crop_width = int((  
        np.linalg.norm(points[0] - points[1])) +  
        np.linalg.norm(points[2] - points[3]))  
    img_crop_height = int((  
        np.linalg.norm(points[0] - points[2])) +  
        np.linalg.norm(points[1] - points[3]))  
    pts_std = np.float32([(0, 0), (img_crop_width, 0),  
        (img_crop_width, img_crop_height),  
        (0, img_crop_height)])  
    M = cv2.getPerspectiveTransform(points, pts_std)  
    dst_img = cv2.warpPerspective(  
        img, (img_crop_width, img_crop_height),  
        M, flags=cv2.INTER_LINEAR,  
        borderMode=cv2.BORDER_REPLICATE)  
    dst_img_height, dst_img_width = dst_img.shape[0:2]  
    if dst_img_height * 2.0 > dst_img_width * 2.0:  
        dst_img = np.rot90(dst_img)  
    return dst_img, points
```

```
def get_minimac_rect_crop(img, points):  
    bounding_box = cv2.minAreaRect(np.array(points).astype(np.int32))  
    points = sorted(list(cv2.bboxPoints(bounding_box)), key=lambda x: x[0])  
    index_a = 0  
    index_b = 1  
    index_c = 2  
    index_d = 3  
    if points[1][1] > points[0][1]:  
        index_a = 1  
        index_b = 0  
        index_c = 3  
        index_d = 2  
    box = [points[index_a], points[index_b], points[index_c], points[index_d]]  
    crop_img, box = get_rotate_crop_image(img, np.array(box))  
    return crop_img, box
```

problem: do paddleOCR xử lý tiếng anh và trung nên box bị cắt sát với chữ, đối với tiếng việt có dấu âm nên box cần phải được cắt dư ra để bao gồm dấu

```
def expand_bounding_box(points, expand_ratio=0.2):  
    x_min = np.min(points[:, 0])  
    x_max = np.max(points[:, 0])  
    y_min = np.min(points[:, 1])  
    y_max = np.max(points[:, 1])  
    width = x_max - x_min  
    height = y_max - y_min  
    x_min_new = x_min - expand_ratio * width  
    x_max_new = x_max + expand_ratio * width  
    y_min_new = y_min - expand_ratio * height  
    y_max_new = y_max + expand_ratio * height  
    new_box = np.array([[x_min_new, y_min_new], [x_max_new, y_min_new],  
        [x_max_new, y_max_new], [x_min_new, y_max_new]])  
    return new_box
```

thêm hàm expand bbox ở thư viện paddleocr