

## ДОДАТОК А

### Код допоміжних функцій

Лістинг А.1 – програмний код функції `run_text_ocr`

```
def run_text_ocr(self, image: np.ndarray):
    result = self.craft_pipeline(image,
link_threshold=0.1)
    sorted_polys, sorted_groups = sort_polys(image,
result, verbose=False, config=self.config)
    box_images = crop_all(image, sorted_polys,
min_rotation_angle=30, padding=(0.2, 0.1))

    valid_box_images = []
    updated_line_lengths = []
    idx = 0
    for line in sorted_groups:
        valid_count = 0
        for _ in line:
            img = box_images[idx]
            idx += 1
            if img.shape[0] > 0 and img.shape[1] > 0:
                valid_box_images.append(img)
                valid_count += 1
        if valid_count > 0:
            updated_line_lengths.append(valid_count)

    ocr_result =
self.ukr_recognizer.recognize_batch(valid_box_images)

    lines = []
    i = 0
    for length in updated_line_lengths:
        line = ocr_result[0][i:i+length]
        lines.append(" ".join(line))
```

## Продовження лістингу А.1

```
        i += length

        recognized_text = " | ".join(lines)
        final_result = self.corrector.fit(recognized_text)
        final_lines = final_result.split("|")
        final_text = final_result.replace(" | ", " ")

        return final_text, final_lines, sorted_groups,
ocr_result[0]
```

## Лістинг А.2 – програмний код функції run\_math\_ocr

```
def run_math_ocr(self, images, valid_boxes):
    math_result = [self.math_recognizer.recognize(img) for
img in images]

    filtered_results = []
    filtered_boxes = []
    for latex, box in zip(math_result, valid_boxes):
        if is_valid_latex(latex):
            filtered_results.append(latex)
            filtered_boxes.append(box)

    return filtered_results, filtered_boxes
```

