

Практична робота № 8

Варіант 13

Дослідження методів комп'ютерного зору

Мета: використовуючи спеціалізовані бібліотеки та мову програмування Python навчитися обробляти зображення за допомогою бібліотеки OpenCV

Хід роботи:

Завдання 8.1: Завантаження зображень та відео в OpenCV

Лістинг:

```
import cv2

def load_image():
    img = cv2.imread('data/makovska.png')
    cv2.imshow('Output', img)
    cv2.waitKey(0)

if __name__ == '__main__':
    load_image()
```



Рис.8.1. Результат виконання

					ДУ «Житомирська політехніка».22.121.13.000 – Лр08						
Змн.	Арк.	№ докум.	Підпис	Дата							
Розроб.		Маковська О.Ю			Звіт з лабораторної роботи			Літ.	Арк.	Аркушів	
Перевір.		Пулеко І. В.								1	10
Керівник								ФІКТ Гр. ІПЗ-19-1[2]			
Н. контр.											
Зав. каф.											

Завдання 8.2: Дослідження перетворень зображення

Лістинг:

```
import cv2
import numpy as np

img = cv2.imread('data/makovska.png')
kernel = np.ones((5, 5), np.uint8)

imgGray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
imgBlur = cv2.GaussianBlur(imgGray, (7, 7), 0)
imgCanny = cv2.Canny(img, 150, 200)
imgDilation = cv2.dilate(imgCanny, kernel, iterations=1)
imgEroded = cv2.erode(imgDilation, kernel, iterations=1)

cv2.imshow('Gray Image', imgGray)
cv2.imshow('Blur Image', imgBlur)
cv2.imshow('Canny Image', imgCanny)
cv2.imshow('Dilation Image', imgDilation)
cv2.imshow('Eroded Image', imgEroded)
cv2.waitKey(0)
```

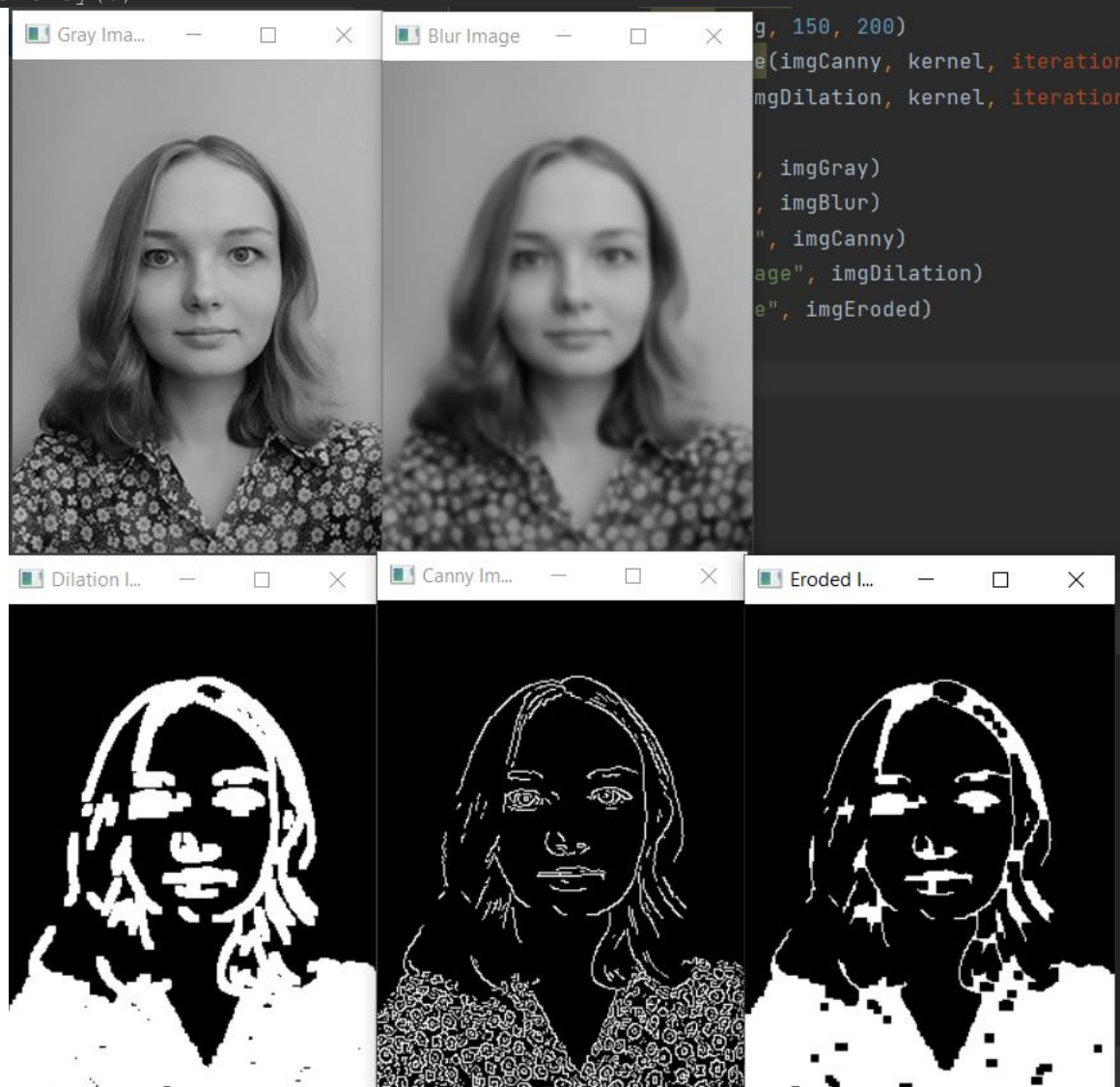


Рис.8.2. Результат виконання

		Маковська О.Ю.			ДУ«Житомирська політехніка».21.121.13.000 – Лр 08	Арк.
		Пулеко І. В.				2
Змн.	Арк.	№ докум.	Підпис	Дата		

Завдання 8.3: Вирізання частини зображення

Лістинг:

```
import cv2

img = cv2.imread('data/makovska.png')
print(img.shape)
imgResize = cv2.resize(img, (1000, 500))
print(imgResize.shape)
imgCropped = img[75:400, 30:350]
cv2.imshow('Image', img)

cv2.imshow('Image Cropped', imgCropped)
cv2.waitKey(0)
```

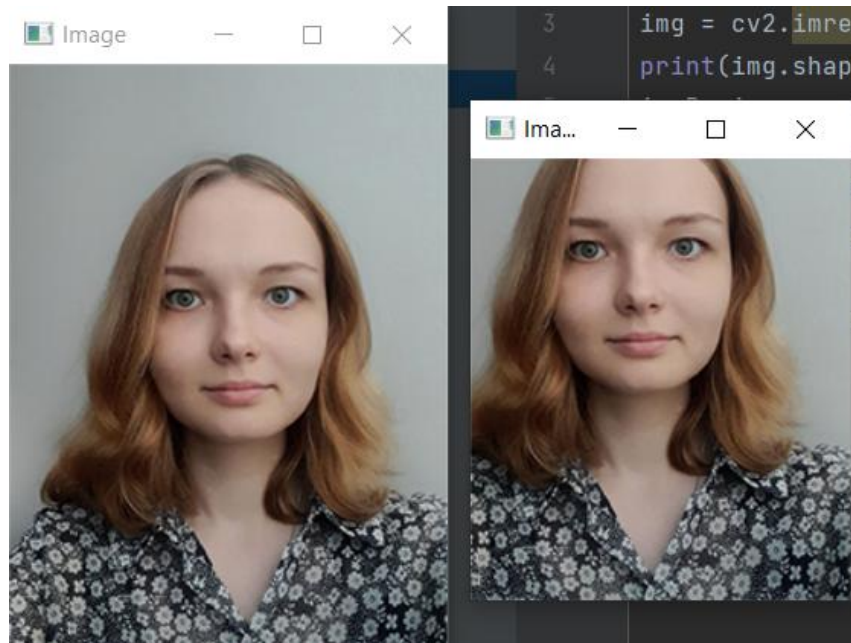


Рис.8.3. Результат виконання

Завдання 8.4: Розпізнавання обличчя на зображенні

Лістинг:

```
import cv2

faceCascade = cv2.CascadeClassifier('data/haarcascade_frontalface_default.xml')
img = cv2.imread('data/makovska.png')
imgGray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

faces = faceCascade.detectMultiScale(imgGray, 1.1, 4)

for (x, y, w, h) in faces:
    cv2.rectangle(img, (x, y), (x + w, y + h), (255, 0, 0), 2)

cv2.imshow('Result', img)
cv2.waitKey(0)
```

		Маковська О.Ю.			ДУ«Житомирська політехніка».21.121.13.000 – Лр 08	Арк.
		Пулеко І.В.				2
Змн.	Арк.	№ докум.	Підпис	Дата		

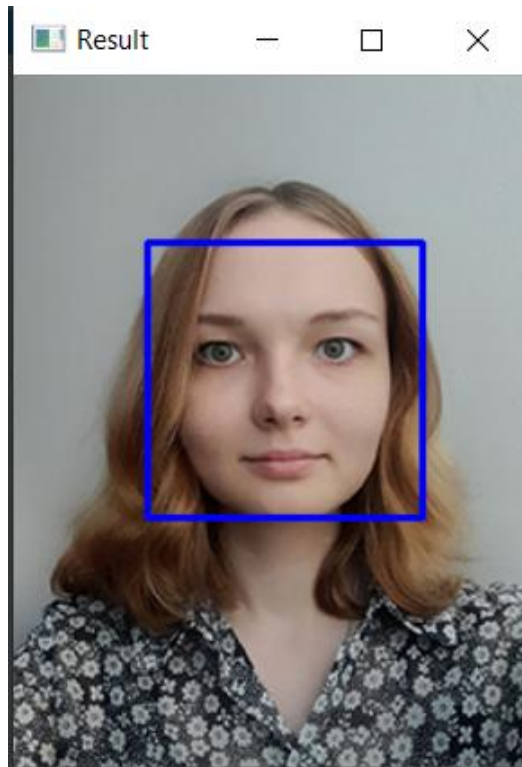


Рис.8.4. Результат виконання

Завдання 8.5: Розпізнавання об'єктів на зображенні за допомогою методів зіставлення шаблонів

Лістинг:

```
img = cv.imread('data/makovska.png', 0)
img2 = img.copy()
template = cv.imread('data/makovska_face.png', 0)
w, h = template.shape[::-1]

methods = ['cv.TM_CCOEFF', 'cv.TM_CCOEFF_NORMED', 'cv.TM_CCORR',
           'cv.TM_CCORR_NORMED', 'cv.TM_SQDIFF', 'cv.TM_SQDIFF_NORMED']
for meth in methods:
    img = img2.copy()
    method = eval(meth)

    res = cv.matchTemplate(img, template, method)
    min_val, max_val, min_loc, max_loc = cv.minMaxLoc(res)

    if method in [cv.TM_SQDIFF, cv.TM_SQDIFF_NORMED]:
        top_left = min_loc
    else:
        top_left = max_loc
    bottom_right = (top_left[0] + w, top_left[1] + h)
    cv.rectangle(img, top_left, bottom_right, 255, 2)
    plt.subplot(121), plt.imshow(res, cmap='gray')
    plt.title('Matching Result'), plt.xticks([], plt.yticks([]))
    plt.subplot(122), plt.imshow(img, cmap='gray')
    plt.title('Detected Point'), plt.xticks([], plt.yticks([]))
    plt.suptitle(meth)
    plt.show()
```

		Маковська О.Ю.			ДУ«Житомирська політехніка».21.121.13.000 – Лр 08	Арк.
		Пулеко І.В.				2
Змн.	Арк.	№ докум.	Підпис	Дата		



Рис.8.5. Результат виконання

		Маковська О.Ю.			ДУ«Житомирська політехніка».21.121.13.000 – Лр 08	Арк.
		Пулеко І. В.				2
Змн.	Арк.	№ докум.	Підпис	Дата		

Завдання 8.6: Сегментація зображення алгоритмом водорозподілу

Лістинг:

```
import numpy as np
import cv2

img = cv2.imread('data/coins_2.jpg')
cv2.imshow('Original', img)
cv2.waitKey(0)

gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
ret, thresh = cv2.threshold(gray, 0, 255, cv2.THRESH_BINARY_INV + cv2.THRESH_OTSU)
cv2.imshow('Threshold', thresh)
cv2.waitKey(0)

kernel = np.ones((3, 3), np.uint8)
opening = cv2.morphologyEx(thresh, cv2.MORPH_OPEN, kernel, iterations=2)

sure_bg = cv2.dilate(opening, kernel, iterations=3)
dist_transform = cv2.distanceTransform(opening, cv2.DIST_L2, 5)
ret, sure_fg = cv2.threshold(dist_transform, 0.7 * dist_transform.max(), 255, 0)

sure_fg = np.uint8(sure_fg)
unknown = cv2.subtract(sure_bg, sure_fg)
cv2.imshow('Coins', opening)
cv2.waitKey(0)

ret, markers = cv2.connectedComponents(sure_fg)
markers = markers + 1
markers[unknown == 255] = 0

markers = cv2.watershed(img, markers)
img[markers == -1] = [255, 0, 0]

cv2.imshow('Markers', img)
cv2.waitKey(0)
```



		Маковська О.Ю.			ДУ«Житомирська політехніка».21.121.13.000 – Лр 08	Арк.
		Пулеко І.В.				3
Змн.	Арк.	№ докум.	Підпис	Дата		

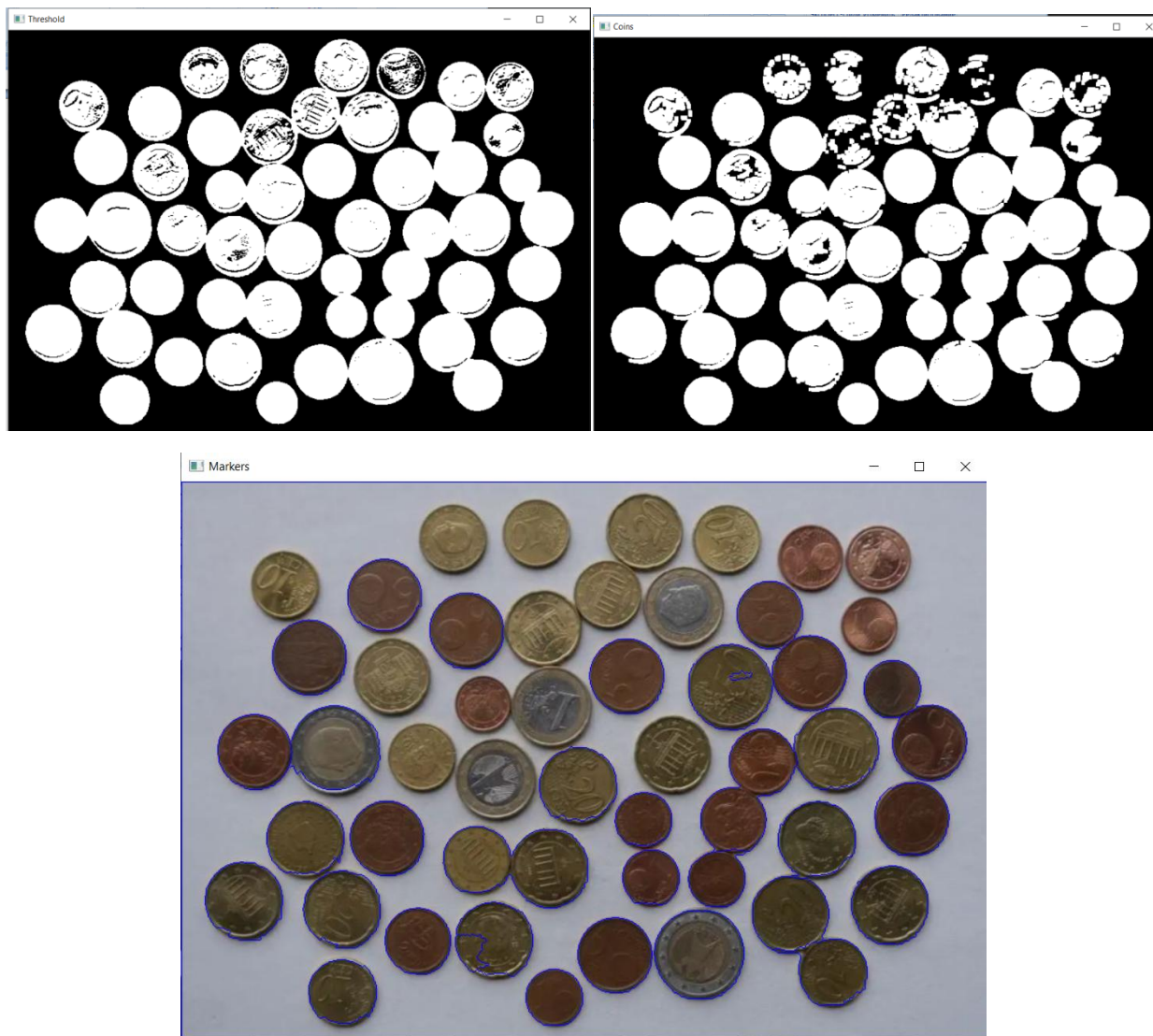


Рис.8.6. Результат виконання

Завдання 8.7: Дослідження перетворень зображення

Лістинг:

```
import cv2
import numpy as np
from scipy import ndimage as ndi
from skimage.feature import peak_local_max
from skimage.segmentation import watershed
import matplotlib.pyplot as plt

img = cv2.imread('coins_2.jpg')
filtered = cv2.pyrMeanShiftFiltering(img, 20, 40)

gray = cv2.cvtColor(filtered, cv2.COLOR_BGR2GRAY)
_, thresh_img = cv2.threshold(gray, 0, 255, cv2.THRESH_BINARY_INV |
cv2.THRESH_OTSU)

contours, _ = cv2.findContours(thresh_img, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
contours_filtered = []
for con in contours:
```

		Маковська О.Ю.			ДУ«Житомирська політехніка».21.121.13.000 – Лр 08	Арк.
		Пулеко І. В.				2
Змн.	Арк.	№ докум.	Підпис	Дата		

```

area = cv2.contourArea(con)
if area < 1000:
    contours_filtered.append(con)

cv2.drawContours(thresh_img, contours_filtered, -1, 255, -1)
dist = ndi.distance_transform_edt(thresh_img)
dist_copy = dist.copy()

local_max = peak_local_max(dist, indices=False, min_distance=20,
labels=thresh_img)
markers = ndi.label(local_max, structure=np.ones((3, 3)))[0]

watershed_img = watershed(-dist, markers, mask=thresh_img)
titles = ['Original image', 'Binary Image', 'Watershed']
images = [img, thresh_img, watershed_img]

fig = plt.gcf()
fig.set_size_inches(8, 6)

for i, img in enumerate(images):
    plt.subplot(3, 1, i + 1)
    plt.imshow(img, 'jet' if i == 2 else 'gray')
    plt.title(titles[i])
    plt.xticks([], plt.yticks([]))

plt.show()

```

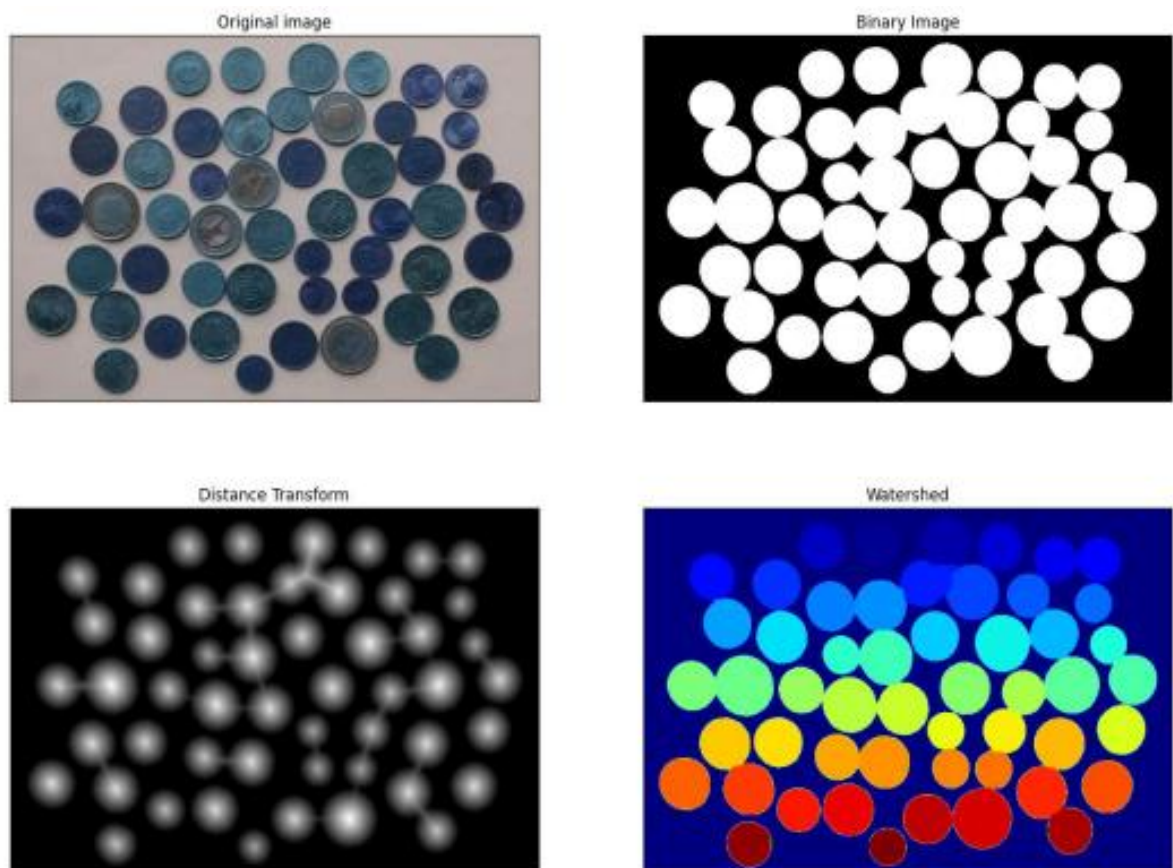


Рис.8.7. Результат виконання

		Маковська О.Ю.			ДУ«Житомирська політехніка».21.121.13.000 – Лр 08	Арк.
		Пулко І. В.				2
Змн.	Арк.	№ докум.	Підпис	Дата		

https://github.com/avrorilka/AI_Python

Висновки: протягом виконання завдань лабораторної роботи було отримано навички з обробки зображення використовуючи бібліотеку OpenCV та мову програмування Python

Під час цих лабораторних завдань ми навчилися використовувати бібліотеки OpenCV для обробки зображень. Ми використали методи cvtColor, GaussianBlur, Canny, dilate та erode для перетворення зображення з одного колірного простору в інший, зменшення шуму та деталізації, виявлення країв і збільшення або зменшення розміру об'єкта на зображенні. Результатом застосування цих методів є зображення в новому колірному просторі зі зниженим шумом і деталізацією, з виділеними краями, а також зі збільшеним або зменшеним розміром об'єкта.

Метод cvtColor використовується для перетворення зображення з одного колірного простору в інший. Його можна використовувати, наприклад, для перетворення зображення з RGB на градації сірого. Результатом застосування цього методу є зображення в новому колірному просторі.

Метод GaussianBlur використовується для зменшення шуму та деталізації зображення. Він застосовує до зображення фільтр Гауса, який розмиває зображення та зменшує кількість шумів і деталей. Результатом застосування цього методу є зображення зі зниженим шумом і деталізацією.

		Маковська О.Ю.			ДУ«Житомирська політехніка».21.121.13.000 – Лр 08	Арк.
		Пулеко І. В.				3
Змн.	Арк.	№ докум.	Підпис	Дата		