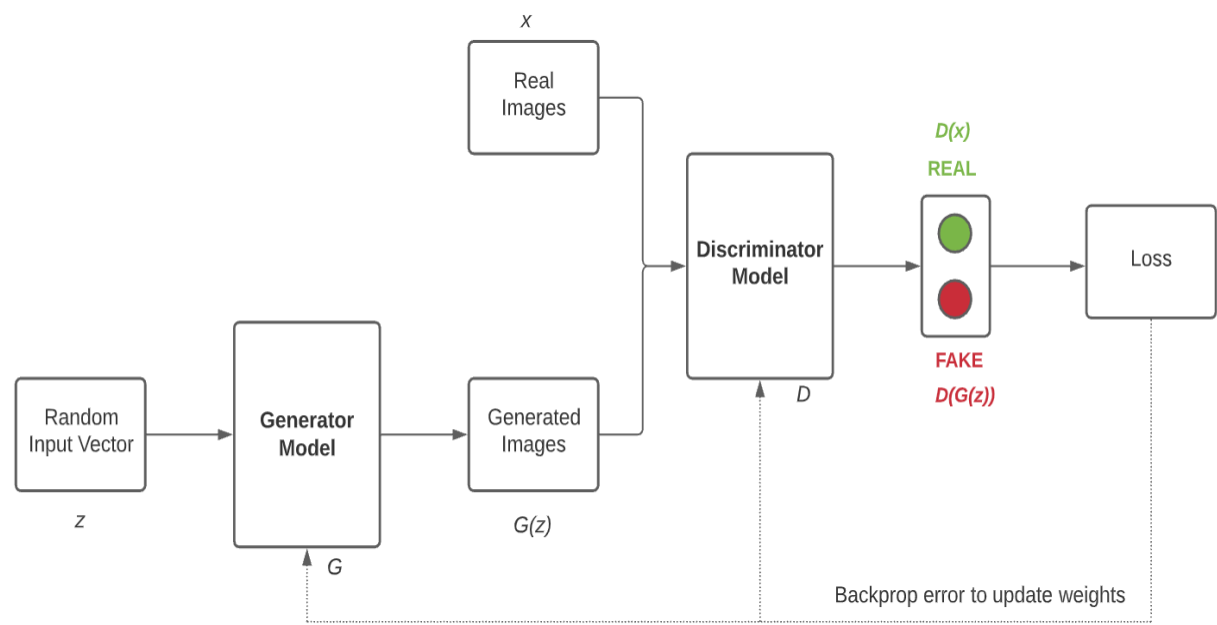


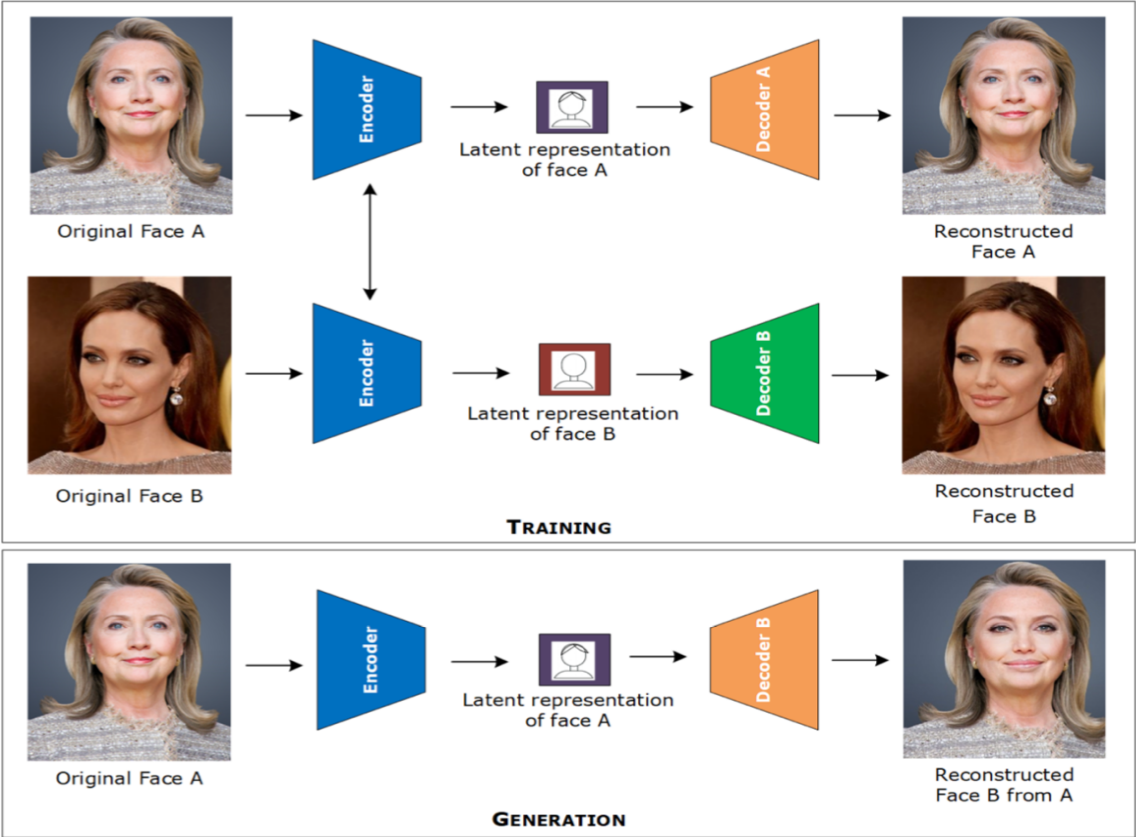
ДОДАТОК А
Графічні матеріали кваліфікаційної роботи

ОРИГІНАЛЬНА АРХІТЕКТУРА ГЗМ



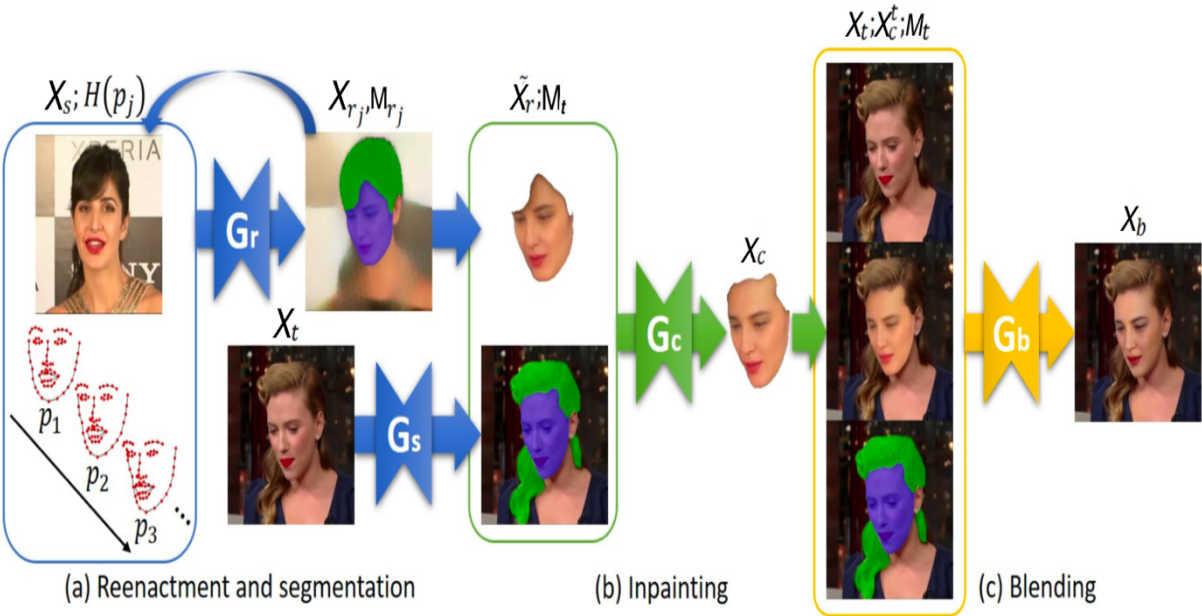
Розроб.	Захаров Р.С.		«Дослідження та розробка методу виявлення підробок зображень та відеоконтенту»		
Перевір.	Калита Н.І.				
Н. Контр.	Калита Н.І.				
				СПРМ-20-1	Лист 1
Завтверд.	Гребеннік І.В.			СТ	Листів 1

СТВОРЕННЯ DEEPFAKE З ВИКОРИСТАННЯМ КОДУВАЛЬНИКІВ ТА
ДЕКОДУВАЛЬНИКІВ



Розроб.	Захаров Р.С.		«Дослідження та розробка методу виявлення підробок зображень та відеоконтенту»	
Перевір.	Калита Н.І.			
Н. Контр.	Калита Н.І.			
			СПРм-20-1	Лист 1
Завтерд.	Гребеннік І.В.		СТ	Листів 1

APXITEKTPYPA FSGAN

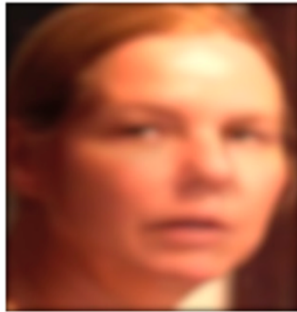


Розроб.	Захаров Р.С.			«Дослідження та розробка методу виявлення підробок зображень та відеоконтенту»	
Перевір.	Калита Н.І.				
Н. Контр.	Калита Н.І.				
				СПРМ-20-1	Лист 1
Завтерд.	Гребеннік І.В.			СТ	Листів 1

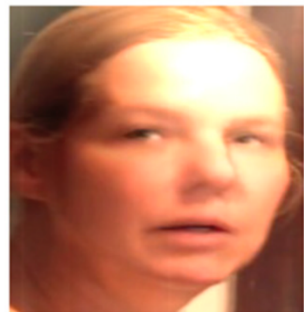
**ПРИКЛАД ЗОБРАЖЕНЬ, ІЗ ЗМІНЕНИМИ ХАРАКТЕРИСТИКАМИ, ЯКІ НЕ
БУЛИ ВИЯВЛЕНІ МЕТОДАМИ ВИЯВЛЕННЯ ПІДРОБОК З
ВИКОРИСТАННЯМ ЗНМ**



а)



б)



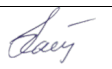
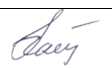
в)



г)

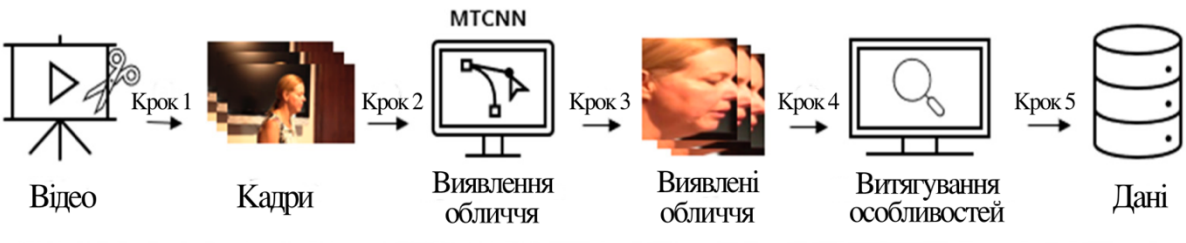


г)

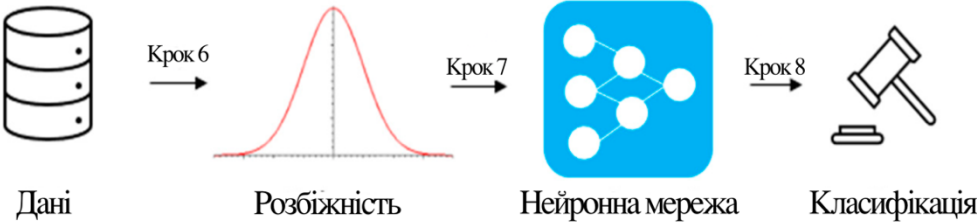
<i>Розроб.</i>	<i>Захаров Р.С.</i>			<i>«Дослідження та розробка методу виявлення підробок зображень та відеоконтенту»</i>	
<i>Перевір.</i>	<i>Калита Н.І.</i>				
<i>Н. Контр.</i>	<i>Калита Н.І.</i>				
				<i>СПРМ-20-1</i>	<i>Лист 1</i>
<i>Завтверд.</i>	<i>Гребеннік І.В.</i>			<i>СТ</i>	<i>Листів 1</i>

АРХІТЕКТУРА ТЕХНОЛОГІЇ ТА АЛГОРИТМ РОЗРОБЛЕНОГО МЕТОДУ
ВИЯВЛЕННЯ ПІДРОБОК МЕДІА-ФАЙЛІВ

Попередня обробка даних



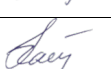
Класифікація



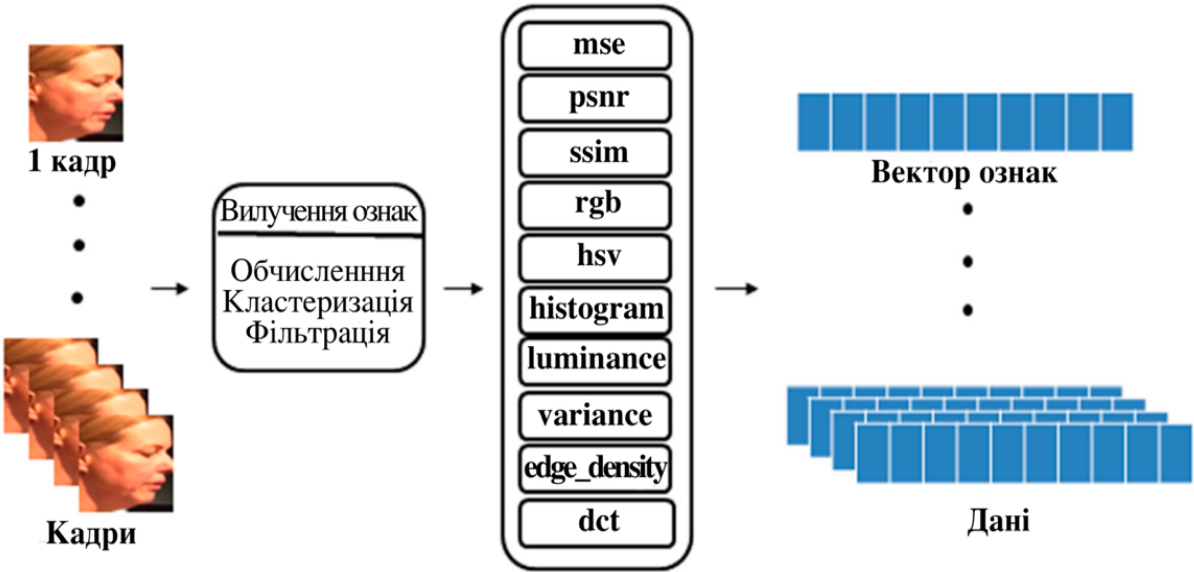
Розроб.	Захаров Р.С.		«Дослідження та розробка методу виявлення підробок зображень та відеоконтенту»		
Перевір.	Калита Н.І.				
Н. Контр.	Калита Н.І.				
				СПРм-20-1	Лист 1
Завтерд.	Гребеннік І.В.			СТ	Листів 1

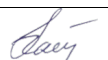
ОТРИМАНІ РЕЗУЛЬТАТИ РОБОТИ МЕРЕЖІ P-NET ПРИ ВИЯВЛЕННІ ОБЛИЧЧЯ НА ЗОБРАЖЕННІ

Kernel Coordinates	Bounding box: x1	Bounding box: y1	Bounding box: x2	Bounding box: y2	Confidence
(0,5)	0.50	0.25	0.80	0.58	0.98
(0,6)	0.45	0.17	0.75	0.49	0.96
(0,7)	0.52	0.08	0.73	0.41	0.94
(1,4)	0.42	0.34	0.67	0.66	0.92
(1,8)	0.40	0.01	0.66	0.32	0.96
(3,5)	0.32	0.22	0.49	0.59	0.88
(3,6)	0.25	0.20	0.52	0.53	0.91
(6,6)	0.01	0.18	0.25	0.50	0.89
(6,8)	0.02	0.00	0.22	0.33	0.90

Розроб.	Захаров Р.С.			«Дослідження та розробка методу виявлення підробок зображень та відеоконтенту»	
Перевір.	Калита Н.І.				
Н. Контр.	Калита Н.І.				
				СПРм-20-1	Лист 1
Завтверд.	Гребеннік І.В.			СТ	Листів 1

ПРОЦЕС ОТРИМАННЯ ОЗНАК КОМП'ЮТЕРНОГО ЗОРУ



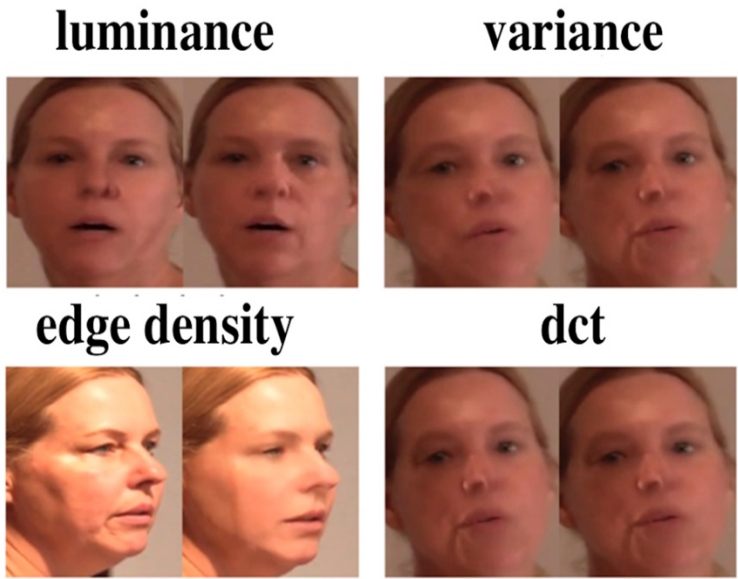
Розроб.	Захаров Р.С.		«Дослідження та розробка методу виявлення підробок зображень та відеоконтенту»		
Перевір.	Калита Н.І.				
Н. Контр.	Калита Н.І.				
				СПРм-20-1	Лист 1
Завтерд.	Гребеннік І.В.			СТ	Листів 1

ЗНАХОДЖЕННЯ ОБЛИЧЧЯ НА ОДНОМУ З КАДРІВ ПІДРОБЛЕНОГО ВІДЕО ЗА ДОПОМОГОЮ ВИКОРИСТАННЯ МЕРЕЖІ МТСNN



Розроб.	Захаров Р.С.			«Дослідження та розробка методу виявлення підробок зображень та відеоконтенту»	
Перевір.	Калита Н.І.				
Н. Контр.	Калита Н.І.				
				СПРм-20-1	Лист 1
Завтверд.	Гребеннік І.В.			СТ	Листів 1

КАДРИ З НАЙБІЛЬШОЮ РОЗБІЖНІСТЮ КОЖНОЇ ОЗНАКИ З ОДНОГО ПІДРОБЛЕНОГО ВІДЕО



Розроб.	Захаров Р.С.			«Дослідження та розробка методу виявлення підробок зображень та відеоконтенту»	
Перевір.	Калита Н.І.				
Н. Контр.	Калита Н.І.				
				СПРМ-20-1	Лист 1
Завтверд.	Гребеннік І.В.			СТ	Листів 1

ДОДАТОК Б
Текст програми

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

«ЗАТВЕРДЖУЮ»

Керівник кваліфікаційної роботи

 проф. Калита Н.І.
(підпис)

«___» _____ 2021 р.

«ДОСЛІДЖЕННЯ ТА РОЗРОБКА МЕТОДУ ВИЯВЛЕННЯ ПІДРОБОК
ЗОБРАЖЕНЬ ТА ВІДЕОКОНТЕНТУ»

Текст програми
ЛИСТ ЗАТВЕРДЖЕННЯ
01.12.01-ЛЗ

УЗГОДЖЕНО:

РОЗРОБИВ:

студент групи СПРм-20-1

Захаров Роман Сергійович

«___» _____ 2021 р.

Затверджено

01.12.01-ЛЗ

**«ДОСЛІДЖЕННЯ ТА РОЗРОБКА МЕТОДУ ВИЯВЛЕННЯ ПІДРОБОК
ЗОБРАЖЕНЬ ТА ВІДЕОКОНТЕНТУ»**

Текст програми

01.12.01

Листів 8

2021

Installation of MTCNN

```
pip install mtcnn
```

Getting all frames from video

```
import cv2
vidcap = cv2.VideoCapture(woman_deepfake.mp4')
success,image = vidcap.read()
count = 0
while success:
    cv2.imwrite("frame%d.jpg" % count, image)      # save frame as
    JPEG file
    success,image = vidcap.read()
    print('Read a new frame: ', success)
    count += 1
```

Saving frames from video

```
import numpy as np
import cv2

cap = cv2.VideoCapture(0)

fourcc = cv2.VideoWriter_fourcc(*'XVID')
out = cv2.VideoWriter('output.avi', fourcc, 20.0, (640,480))

while(cap.isOpened()):
    ret, frame = cap.read()
    if ret==True:
        frame = cv2.flip(frame,0)

        out.write(frame)

        cv2.imshow('frame',frame)
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break
    else:
        break
cap.release()

out.release()

cv2.destroyAllWindows()
```

Face Detection

```
from mtcnn import MTCNN
import cv2

detector = MTCNN()

img = cv2.imread("woman_deep.jpg")
detections = detector.detect_faces(img)

for detection in detections:
    score = detection["confidence"]
    if score & > 0.90:
        x, y, w, h = detection["box"]
        detected_face = img[int(y):int(y+h), int(x):int(x+w)]
```

Drawing a box around faces

```
draw an image with detected objects
def draw_facebox(filename, result_list):

    load the image
    data = plt.imread(filename)
    plot the image
    plt.imshow(data)

    get the context for drawing boxes
    ax = plt.gca()
    plot each box
    for result in result_list:

        get coordinates
        x, y, width, height = result['box']

        create the shape
        rect = plt.Rectangle((x, y), width, height, fill=False,
                             color='green')
        draw the box
        ax.add_patch(rect)

    show the plot
    plt.show()
    filename = 'woman_deep.jpg'  filename is defined above, otherwise
    uncomment
```

```
load image from file
```

```

pixels = plt.imread(filename)  defined above, otherwise uncomment

detector is defined above, otherwise uncomment
detector = mtcnn.MTCNN()
detect faces in the image

faces = detector.detect_faces(pixels)
display faces on the original image
draw_facebox(filename, faces)

Displaying eyes, mouth and nose around faces

draw the dots
for key, value in result['keypoints'].items():
    create and draw dot
    dot = plt.Circle(value, radius=20, color='orange')
    ax.add_patch(dot)

draw an image with detected objects
def draw_facebox(filename, result_list):

    load the image
    data = plt.imread(filename)
    plot the image
    plt.imshow(data)

    get the context for drawing boxes
    ax = plt.gca()
    plot each box

    for result in result_list:

        get coordinates
        x, y, width, height = result['box']

        create the shape
        rect = plt.Rectangle((x, y), width, height, fill=False,
                              color='orange')

        draw the box
        ax.add_patch(rect)

        draw the dots
        for key, value in result['keypoints'].items():

```

```

create and draw dot
dot = plt.Circle(value, radius=20, color='red')
ax.add_patch(dot)

show the plot
plt.show()

filename = 'woman_deep.jpg'  filename is defined above, otherwise
uncomment

load image from file
pixels = plt.imread(filename)  defined above, otherwise uncomment
detector is defined above, otherwise uncomment

detector = mtcnn.MTCNN()
detect faces in the image
faces = detector.detect_faces(pixels)
display faces on the original image
draw_facebox(filename, faces)

```

Advanced Speed up of MTCNN

```

from facenet_pytorch import MTCNN
from PIL import Image
import torch
from imutils.video import FileVideoStream
import cv2
import time
import glob
from tqdm.notebook import tqdm
device = 'cuda' if torch.cuda.is_available() else 'cpu'
filenames = ["woman_deep.jpg", "woman_deep1.jpg"]

```

Defining the extractor

```

from facenet_pytorch import MTCNN
from PIL import Image

import torch
from imutils.video import FileVideoStream

import cv2

import time

```

```

import glob

from tqdm.notebook import tqdm

device = 'cuda' if torch.cuda.is_available() else 'cpu'

filenames = ["woman_deep.jpg", "woman_deep1.jpg"]

```

Face extraction script

```

def run_detection(fast_mtcnn, filenames):
    frames = []
    frames_processed = 0
    faces_detected = 0

    batch_size = 60
    start = time.time()

    for filename in tqdm(filenames):

        v_cap = FileVideoStream(filename).start()
        v_len = int(v_cap.stream.get(cv2.CAP_PROP_FRAME_COUNT))

        for j in range(v_len):

            frame = v_cap.read()
            frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
            frames.append(frame)

            if len(frames) >= batch_size or j == v_len - 1:

                faces = fast_mtcnn(frames)

                frames_processed += len(frames)
                faces_detected += len(faces)
                frames = []

```

```

print(
f'Frames per second: {frames_processed / (time.time() -
start):.3f}',',
f'faces detected: {faces_detected}\r',
end=''
)

v_cap.stop()

run_detection(fast_mtcnn, filenames)

```

Histogram Calculation using OpenCV

```

img = cv.imread('woman_deep.jpg',0)
hist = cv.calcHist([img],[0],None,[256],[0,256])

import numpy as np
import cv2 as cv
from matplotlib import pyplot as plt

img = cv.imread('woman_deep.jpg')
color = ('b','g','r')

for i,col in enumerate(color):
    histr = cv.calcHist([img],[i],None,[256],[0,256])
    plt.plot(histr,color = col)
    plt.xlim([0,256])

plt.show()

```

HSV Calculation using OpenCV

```

red = np.uint8([[255,0,0 ]])
hsv_red = cv.cvtColor(red,cv.COLOR_BGR2HSV)
print( hsv_red )

green = np.uint8([[0,255,0 ]])
hsv_green = cv.cvtColor(green,cv.COLOR_BGR2HSV)
print( hsv_green )

blue = np.uint8([[0,0, 255 ]])
hsv_blue = cv.cvtColor(blue,cv.COLOR_BGR2HSV)
print( hsv_blue )

```

Luminance Calculation using OpenCV

```
import cv2
lum = cv2.imread('woman_deep.jpg',cv2.IMREAD_GRAYSCALE)
```

Variance Calculation using Python

```
import numpy as np
import time
np.random.seed(1234)

img = (np.random.rand(200, 200)*256).astype(np.uint8)

def sliding_window(a, window, axis=-1):
    shape = list(a.shape) + [window]
    shape[axis] -= window - 1
    if shape[axis] < 0:
        raise ValueError("Array too small")
    strides = a.strides + (a.strides[axis],)
    return np.lib.stride_tricks.as_strided(a, shape=shape,
strides=strides)

def sliding_img_var(img, window):
    if window <= 0:
        raise ValueError("invalid window size")
    buf = sliding_window(img, 2*window, 0)
    buf = sliding_window(buf, 2*window, 1)
    out = np.zeros(img.shape, dtype=np.float32)
    np.var(buf[:-1,:-1], axis=(-1,-2), out=out[window:-
window>window,window:-window])
    return out

def looping_img_var(im, w):
    nx, ny = im.shape
    varianceMatrix = np.zeros(im.shape, np.float32)
    for i in range(w,nx-w):
        for j in range(w,ny-w):
            sampleframe = im[j-w:j+w, i-w:i+w]
            variance = np.var(sampleframe)
            varianceMatrix[j][i] = variance
    return varianceMatrix

np.set_printoptions(linewidth=1000, edgeitems=5)
start = time.time()
print(sliding_img_var(img, 1))
time_sliding = time.time() - start
start = time.time()
print(looping_img_var(img, 1))
time_looping = time.time() - start
print("duration: sliding: {0} s, looping: {1}
s".format(time_sliding, time_looping))
```

№	Позначення			Найменування		Примітка	
1.				Пояснювальна записка		87 с.	
2.	01.12.01			Текст програми		8 с.	
				Графічні документи		9 с.	
3.				Оригінальна архітектура ГЗМ		1 с.	
4.				Створення DeepFake з використанням кодувальників та декодувальників		1 с.	
5.				Архітектура FSGAN		1 с.	
6.				Приклад зображень, із зміненими характеристиками, які не були виявлені методами виявлення підробок з використанням ЗНМ		1 с.	
7.				Архітектура технології та алгоритм розробленого методу виявлення підробок медіа-файлів		1 с.	
8.				Отримані результати роботи мережі P-Net при виявленні обличчя на зображенні		1 с.	
9.				Процес отримання ознак комп'ютерного зору		1 с.	
10.				Знаходження обличчя на одному з кадрів підробленого відео за допомогою використання мережі MTCNN		1 с.	
11.				Кадри з найбільшою розбіжністю кожної ознаки з одного підробленого відео		1 с.	
Змін.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Захаров Р.С..					
Перевірила		Калита Н.І.			«Дослідження та розробка методу виявлення підробок зображень та відеоконтенту»	Аркуш	Аркушів
Н. Контр.		Калита Н.І..				1	1
Затвер.		Гребеннік І.В.				ХНУРЕ Кафедра СТ	