ДОДАТОК А

Програмна реалізація моделі нейронної мережі

```
def build_hybrid_unet():
    base = tf.keras.applications.EfficientNetB0(include_top=False, weights='imagenet', input_shape=(256,256,3))

    skips = [
        base.get_layer('block1a_activation').output,
        base.get_layer('block2a_activation').output,
        base.get_layer('block3a_activation').output,
        base.get_layer('block4a_activation').output
    ]
    x = base.output

    clf = layers.GlobalAveragePooling2D()(x)
    clf = layers.Dropout(0.5)(clf)
    clf_out = layers.Dense(3, activation='softmax')(clf)


    for skip in reversed(skips):
        x = layers.UpSampling2D()(x)
        x = layers.Concatenate()([x, skip])
        x = layers.Conv2D(64, 3, padding='same', activation='relu')(x)
        x = layers.Conv2D(64, 3, padding='same', activation='relu')(x)

    x = layers.UpSampling2D()(x)
    seg_out = layers.Conv2D(1, 1, activation='sigmoid')(x)

    return keras.Model(base.input, [seg_out, clf_out])

model = build_hybrid_unet()
```

Рисунок А.1 – Модель нейронної мережі

```python
def predict(image_path, threshold=0.5):

    h, w = img.shape
    img_resized = cv2.resize(img, (256, 256))
    img_input = np.stack([img_resized, img_resized, img_resized], axis=-1) / 255.0
    img_input = np.expand_dims(img_input, axis=0)

    seg_pred, cls_pred = model.predict(img_input)
    seg_pred = seg_pred[0, :, :, 0]
    cls_pred = cls_pred[0]

    seg_bin = (seg_pred > threshold).astype(np.uint8) * 255
    seg_bin = cv2.resize(seg_bin, (w, h), interpolation=cv2.INTER_NEAREST)

    ys, xs = np.nonzero(seg_bin)

    if len(xs) == 0:
        return "normal", 1.0, None, None, None

    x_min, x_max = xs.min(), xs.max()
    y_min, y_max = ys.min(), ys.max()

    boxed = cv2.cvtColor(img, cv2.COLOR_GRAY2BGR)
    cv2.rectangle(boxed, (x_min, y_min), (x_max, y_max), (0, 255, 0), 2)

    class_id = np.argmax(cls_pred)
    class_name = ["benign", "malignant", "normal"][class_id]
    prob = cls_pred[class_id]

    img_bgr = cv2.cvtColor(img, cv2.COLOR_GRAY2BGR)
    seg_bin_bgr = cv2.cvtColor(seg_bin, cv2.COLOR_GRAY2BGR)

    text = f"{class_name} ({prob:.2f})"
    cv2.putText(boxed, text, (10, 30), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2, cv2.LINE_AA)

    combined = np.hstack([img_bgr, seg_bin_bgr, boxed])

    base_name = os.path.basename(image_path)
    mask_full = os.path.join(settings.MEDIA_ROOT, 'segmented', 'mask', base_name)
    boxed_full = os.path.join(settings.MEDIA_ROOT, 'segmented', 'boxed', base_name)
    vis_full   = os.path.join(settings.MEDIA_ROOT, 'segmented', 'visualization', base_name)


    os.makedirs(os.path.dirname(mask_full), exist_ok=True)
    os.makedirs(os.path.dirname(boxed_full), exist_ok=True)
    os.makedirs(os.path.dirname(vis_full), exist_ok=True)

    Image.fromarray(seg_bin).save(mask_full)
    Image.fromarray(boxed).save(boxed_full)
    Image.fromarray(combined).save(vis_full)

    mask_rel = f"segmented/mask/{base_name}"
    boxed_rel = f"segmented/boxed/{base_name}"
    vis_rel = f"segmented/visualization/{base_name}"

    return class_name, prob, (mask_full, mask_rel), (boxed_full, boxed_rel), (vis_full, vis_rel)
```

Рисунок А.2 – Функція передобробки даних

```
CREATE TABLE IF NOT EXISTS public.myapp1_recognitions
(
    id integer NOT NULL GENERATED BY DEFAULT AS IDENTITY ( INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 2147483647 CACHE 1 ),
    "Image_id_id" integer NOT NULL,
    "User_id_id" bigint NOT NULL,
    "Boxed_image" character varying(100) COLLATE pg_catalog."default",
    "Mask_image" character varying(100) COLLATE pg_catalog."default",
    "Visualization_image" character varying(100) COLLATE pg_catalog."default",
    folder_id bigint,
    CONSTRAINT myapp1_recognitions_pkey PRIMARY KEY (id),
    CONSTRAINT "myapp1_recognitions_Image_id_id_5ebbaf89_fk_myapp1_images_id" FOREIGN KEY ("Image_id_id")
        REFERENCES public.myapp1_images (id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        DEFERRABLE INITIALLY DEFERRED,
    CONSTRAINT "myapp1_recognitions_User_id_id_8b1d47af_fk_myapp1_profile_id" FOREIGN KEY ("User_id_id")
        REFERENCES public.myapp1_profile (id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        DEFERRABLE INITIALLY DEFERRED,
    CONSTRAINT myapp1_recognitions_folder_id_f763f920_fk_myapp1_pa FOREIGN KEY (folder_id)
        REFERENCES public.myapp1_patientfolder (id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        DEFERRABLE INITIALLY DEFERRED
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS public.myapp1_recognitions
    OWNER to postgres;
-- Index: myapp1_recognitions_Image_id_id_5ebbaf89

-- DROP INDEX IF EXISTS public."myapp1_recognitions_Image_id_id_5ebbaf89";

CREATE INDEX IF NOT EXISTS "myapp1_recognitions_Image_id_id_5ebbaf89"
    ON public.myapp1_recognitions USING btree
    ("Image_id_id" ASC NULLS LAST)
    WITH (fillfactor=100, deduplicate_items=True)
    TABLESPACE pg_default;
-- Index: myapp1_recognitions_User_id_id_8b1d47af

-- DROP INDEX IF EXISTS public."myapp1_recognitions_User_id_id_8b1d47af";

CREATE INDEX IF NOT EXISTS "myapp1_recognitions_User_id_id_8b1d47af"
    ON public.myapp1_recognitions USING btree
    ("User_id_id" ASC NULLS LAST)
    WITH (fillfactor=100, deduplicate_items=True)
    TABLESPACE pg_default;
-- Index: myapp1_recognitions_folder_id_f763f920

-- DROP INDEX IF EXISTS public.myapp1_recognitions_folder_id_f763f920;

CREATE INDEX IF NOT EXISTS myapp1_recognitions_folder_id_f763f920
    ON public.myapp1_recognitions USING btree
    (folder_id ASC NULLS LAST)
    WITH (fillfactor=100, deduplicate_items=True)
    TABLESPACE pg_default;
```

Рисунок А.3 – SQL-скрипт створення головної таблиці «Recognition»

ДОДАТОК Б

Відомість кваліфікаційної роботи

Позначення	Найменування	Дод. відомості
	Текстові документи	
1.	Пояснювальна записка	69 с.
	Інші документи	
2.	Презентаційні матеріали	!8 плакатів

.	.	Прізвище та ініціали.	Підп.	Дата			
Розробив		Михайлов С.В.			Застосування глибоких згорткових мереж для автоматичного аналізу медичних зображень	Шифр групи	Код напр./спец.
Перевірив		Дейнеко А.О.				ДСм-24-1	122
Н.контр.		Малєєва І.А.				ХНУРЕ кафедра ШІ	
Затв.		Чала Л.Е.					