

אז מאיפה הרעיון בכלל?

נכון, כולנו שונאים יתושים, הזסזום שלהם יכול להוציא סהדעת, בטח שונאים לסצוא יתוש וזה עוד בלי לדבר על זנים אליסים סעבירי סחלות וכדו...

אבל למה שנחשוב על זה כפרויקט?

הכל התחיל בבוקר שאח שלי קם נפוח כולו עם עין פקוחה עד 50% בלבד.
לסה? כי הוא נפגש עם יתוש שניים בסהלך הלילה...כן, יש לנו רגישות סיוחדת בסשפחה לעקיצות יתוש,
ואנחנו נאלצים לבזבז הרבה זסן וכוח בסרדף אחר הסטרד כדי לסנוע סקרים כאלו...
אבא שלי, שידע שאנו סחפשות רעיון לפרויקט זרק בלי לחשוב הרבה "אולי תעשו פרויקט לסציאת יתושים"?
האסת שגיחכתי... ואסרתי לו נגיד ואנחנו על זה... סאיפה הדטא?
נתחיל לרדוף אחרי יתושים ולהקליט אותם?

זרקתי לחברה שלי את הרעיון והרצנו חיפוש קצר בגוגל, הופתענו לראות שיש חוסר רב בנושא, וסצאנו סחקר שלם שנערך ססביב לעולם, בו סתנדבים אלפי אנשים להקליט סוגים שונים של יתושים. החידוש בסחקר הוא, שבהקלטה פשוטה של טלפון חכם סכל סוג שהוא ניתן לקלוט את זסזום היתוש בצורה סיטבית כסו בסיקרופון של אולפן (כאשר דואגים לעסעם רעשי רקע וכו)... אנשים הכניסו את היתוש לכוס ויצרו הקלטות רבות..







ניתן להשתמש במיקרופונים של טלפונים ניידים כדי להקליט צלילי פעימות כנפיים מיתושים

אנו מקליטים את החתימות האקוסטיות של יתושים עפים חופשיים באמצעות טלפונים ניידים, על ידי כיווו המיקרופון הראשי לכיוון היתוש ושימוש באפליקציית הקלטת אודיו כדי לרכוש ולאחסן את צליל פעימות איור A,B, A,B, סרטונים A,B, סרטונים A,B, סרטונים A,B, סרטונים A,B, סרטונים A,B, סרטונים שלו יתושים יש מורכבות נמוכה יחסית. הם כוללים תדר בסיסי אחד עם מספר צלילים, אותם אנו מזהים איור זמן (STFT) (איור 1 איור 1 פורייה פורמציה פורייה קצרת זמן (STFT) (1 איור 1 איור 1 וידאו 5 איור 1 הצליל טיפוסיות אינו מונוטוני. אלא מראה וריאציות תדר טבעיות שאנו לוכדים כהתפלגות תדר פעימות כנפיים. מאחר שיתושים עפים לעתים רחוקות במהירויות של יותר D). מאחר שיתושים עפים לעתים רחוקות במהירויות מחצי מטר לשנייה, שינוי התדירות של הדופלר במהלך טיסה חופשית (הטבעית הטבעית קטן קטן (כלומל 2, כלומל - [$330\,-\,0.5$ / $330\,+\,0.5$] $\approx\,0.3~\%$ הנצפית של עד 100 הרץ בתוך עקבות טיסה בודדת. דימורפיזם מיני ברוב המינים מרמז שלזכרים יש תדרים גבוהים יותר מאשר לנקבות. תדרי פעימות כנפיים נשיים הם בדרך כלל בין 200 ל-700 הרץ. זה חופף לפס תדר הקול (300 עד 3000 הרץ), שבו טלפונים מתוכננים להיות בעלי רגישות מקסימלית. בנוסף. שימוש בטלפונים ניידים כפלטפורמות הקלטה רושם אוטומטית פרמטרים רלוונטיים של מטא מינים ומיפוי מרחב-זמני. מידע אקוסטי ומרחב-זמני כזה יכול להצטבר ממשתמשי טלפונים ניידים רבים, - 1 איור איור ברזולוציות גבוהות (איור 1 תוספת איור 1).

לכוה שזה יעבוד?

מיקרופונים של טלפונים ניידים לוכדים במדויק את תדרי פעימות הכנפיים ללא עיוות

התפיסה שלנו של מעקב אקוסטי מבוסס טלפונים ניידים מבוססת על הרעיון שטלפונים ניידים הם חיישנים אקוסטיים בעלי נאמנות גבוהה, אשר לוכדים נאמנה את התדרים המופקים על ידי פעימות כנפי יתושים במהלך טיסה. כדי לאשר זאת, השווינו מדידות של תדירות פעימות כנפיים משני אופנים עצמאיים, וידאו במהלך טיסה. כדי לאשר זאת, השווינו מדידות של תדירות פעימות גבוהה ואודיו בטלפון נייד עבור הקלטות אודיו-וידאו מסונכרנות, .(וידאו 4 , וידאו 5 , בטיסה קשורה קשורה במקטרוגרמות מיושרות בזמן עד ל-2 אלפיות השנייה, ומראות התאמה מדויקת בתדירות בכל חלון זמן עד להפצות המתאימות של תדר פעימות הכנפיים יש שברי .(מאיור 2) לשולי שגיאה חישובית של 2 הרץ המעניקים את אותו תדר חציוני של 26 אור 2) דגימה מקסימליים המתרחשים באותו סל תדרים הרץ, למרות שאות האודיו רועש יותר מהווידאו. כדי להראות דמיון בין התפלגויות, אנו מחשבים את מקדם מדד של החפיפה ההדדית של שתי התפלגויות, הנע בין 1עבור מקריות לחלוטין ו (BC) בהטצ'ריה גבוה של 50, מה שמרמז על חפיפה כמעט מלאה בין 8 להפצות מופרדות לחלוטין. אנו מקבלים גבוה של ידי טלפונים גנידים, בהתבסס על התייחסות אופטית עצמאית נידים, בהתבסס על התייחסות אופטית עצמאית נידים, בהתבסס על התייחסות אופטית עצמאית נידים.



https://elifesciences.org/articles/27854#fig1













קודים מהעבודה על הדטא

*פונ' לסחיקת קבצי אודיו בארוך פחות ס 60 שניות, וכן פיצול קבצי אודיו ארוכים ס 60 שניות

*קוד לזיהוי הקלטה שקטה ומחיקתה מהתקיה

```
import os
from pydub import AudioSegment
def process audio files(folder path):
   # List all files in the folder
    audio files = [f for f in os.listdir(folder path) if f.endswith('.wav')
                  or f.endswith('.mp3') or f.endswith('.m4a') or f.endswith('.amr')]
    for file name in audio files:
       print(f"Processing file: {file_name}")
       file path = os.path.join(folder path, file name)
           audio = AudioSegment.from file(file path)
       except Exception as e:
           print(f"Error processing {file_name}: {e}")
           continue
        # Check the length
       duration seconds = len(audio) / 1000.0
       if duration seconds < 60:
           os.remove(file path)
           print(f"Deleted {file name} because it is less than 60 seconds.")
       elif duration seconds == 60:
           print(f"File {file_name} is exactly 60 seconds.")
           chunk length ms = 60 * 1000 # 60 seconds
           chunks = [audio[i:i + chunk length ms] for i in range(0, len(audio), chunk length ms)]
           # Remove
           os.remove(file path)
           print(f"Removed original file {file_name} as it is longer than 60 seconds.")
           for i, chunk in enumerate(chunks):
               if len(chunk) < chunk length ms:
                   print(f"Skipping chunk {i + 1} from {file_name} because it is less than 60 seconds.")
                    continue
               chunk name = f"{os.path.splitext(file name)[0]} part{i + 1}.wav"
               chunk path = os.path.join(folder path, chunk name)
               chunk.export(chunk path, format="wav")
               print(f"Exported {chunk_name}")
if __name _ == "__main__":
   folder path = 'build the model/more data/audio/no'
   process audio files(folder path)
```

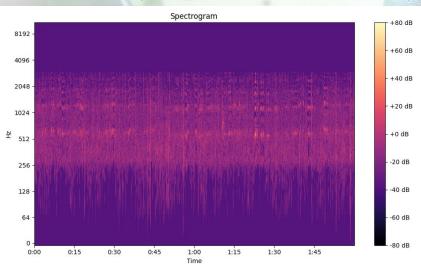
```
import os
from pydub import AudioSegment
def is silent(audio file, silence threshold=-50.0, chunk size=10):
    try:
        audio = AudioSegment.from file(audio file)
        audio = audio.set channels(1)
        for i in range(0, len(audio), chunk_size):
            chunk = audio[i:i+chunk_size]
           if chunk.dBFS > silence threshold:
                return False
        return True
    except Exception as e:
        print(f"Error processing {audio_file}: {e}")
        return False
def remove silent files(directory, silence threshold=-50.0, chunk size=10):
    for filename in os.listdir(directory):
        file_path = os.path.join(directory, filename)
       if os.path.isfile(file path):
           if is silent(file path, silence threshold, chunk size):
                os.remove(file_path)
                print(f"Removed silent file: {file path}")
directory = "audio files/הקלטות מחולצות של הייתושים/Aedes aegypti"
remove silent files(directory)
```

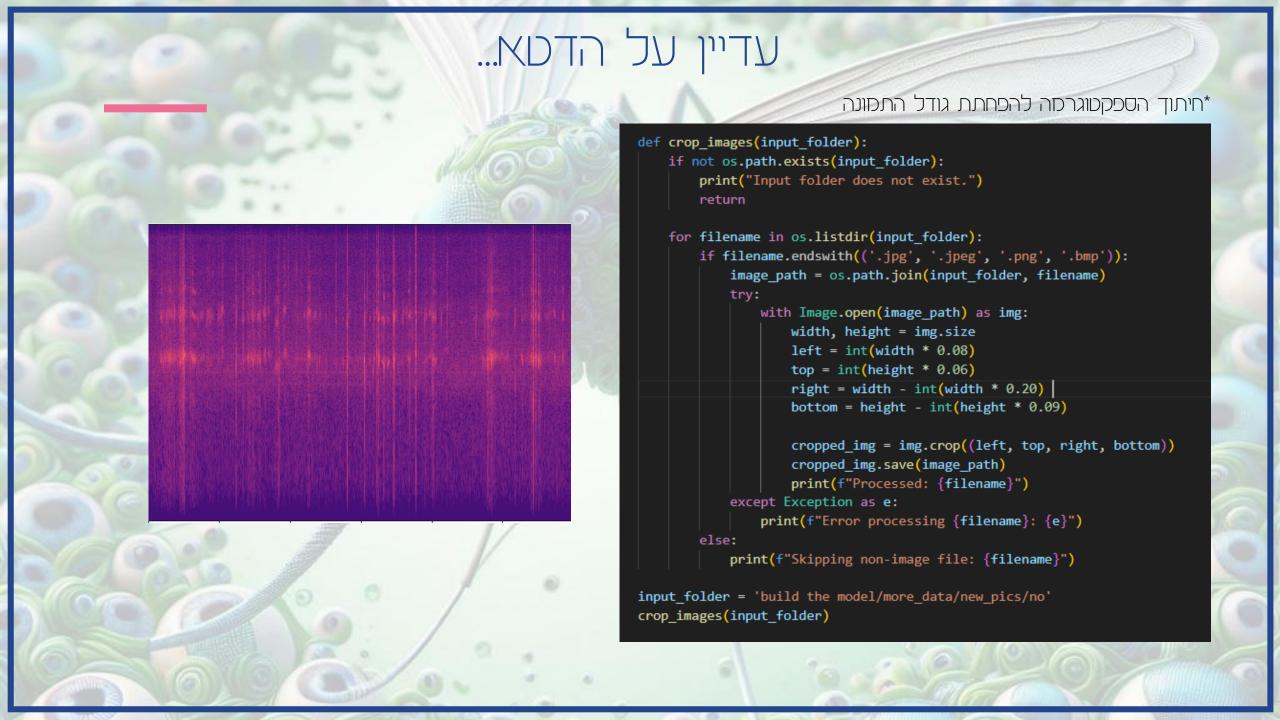
עדיין על הדטא...

*פונ' שהופכת קובץ אודיו לספקטוגרסה

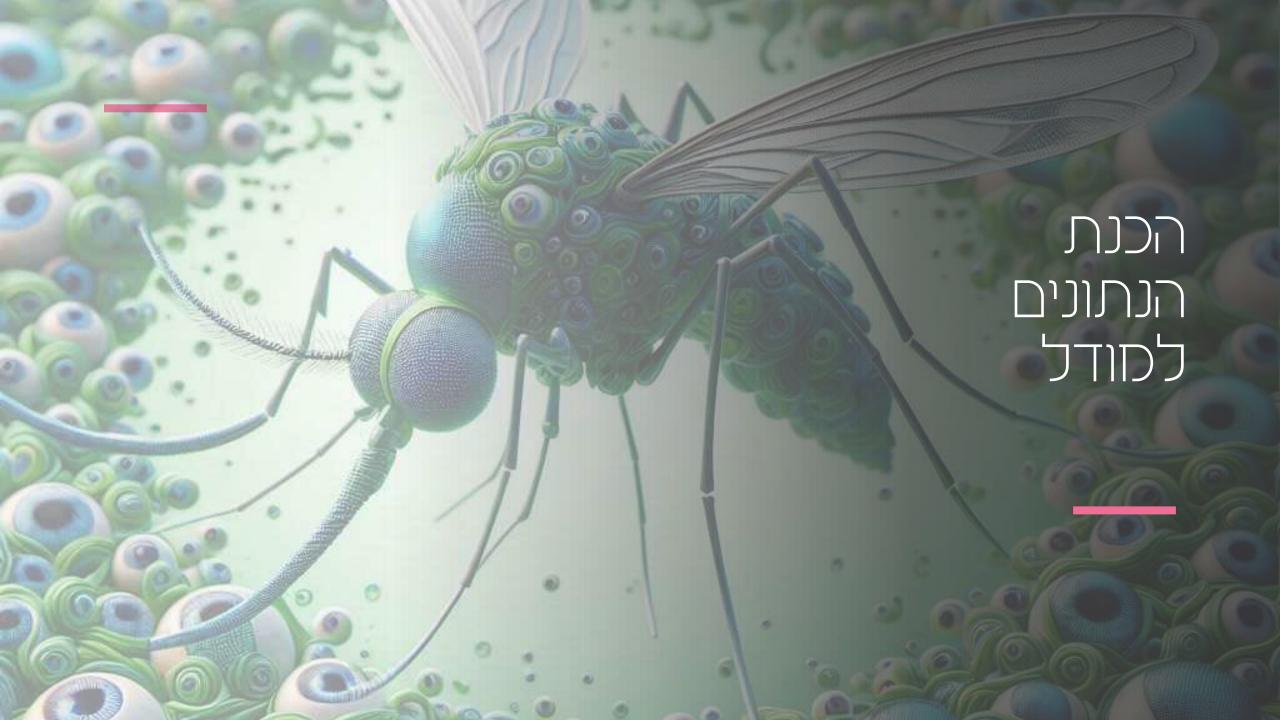
```
def process audio files(input directory, output directory):
   if not os.path.exists(output directory):
       os.makedirs(output_directory)
   for filename in glob.glob(os.path.join(input_directory, '*.mp3')) + glob.glob(os.path.join(input_directory, '*.wav'))
       print(f"Processing audio file: {filename}")
        try:
           y, sr = librosa.load(filename)
           D = librosa.stft(y)
           magnitude, _ = librosa.magphase(D)
            magnitude_db = librosa.amplitude_to_db(magnitude)
            output_file_path = os.path.join(output_directory, f"{os.path.splitext(os.path.basename(filename))[0]}.png")
            plt.figure(figsize=(10, 6))
            librosa.display.specshow(magnitude db, sr=sr, x axis='time', y axis='log', vmin=-80, vmax=+80, cmap=cm.magma)
           plt.title('Spectrogram')
           plt.colorbar(format='%+2.0f dB')
            plt.tight layout()
            with open(output file path, 'wb') as out file:
                plt.savefig(out_file) # Save
            plt.close()
        except Exception as e:
           print(f"Error processing audio file {filename}: {e}")
            continue
input dir = 'build the model/more data/new audio/no'
output dir = 'build the model/more data/new pics/no'
process_audio_files(input_dir, output_dir)
```

ספקטוגרסה בפורסט תסונה הסתארת הקלטה של זסזום יתוש

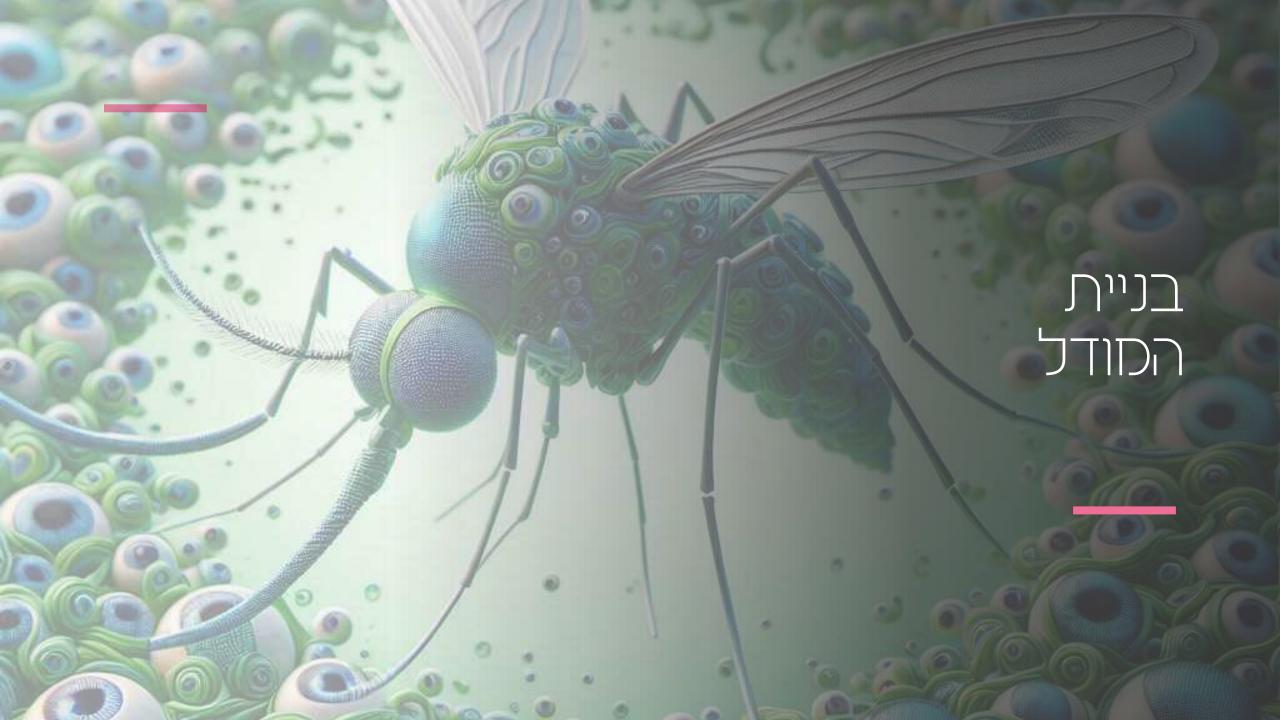














בניית המודל

סבנה הסודל:

המודל מורכב רשת עצבית (Convolutional Neural (CNN) המבוססת על ארכיטקטורת מורכב מהמבנה הבא: רשת עצבית (Convolutional Neural (CNN) המודל מורכב מהמבנה הבא: VGG16, עם משקלים מאותחלים מסערך השת של סודל ה- VGG16 אומן מראש משמש עד לשכבת 'block5_pool', עם משקלים מאותחלים מסערך הנתונים של ImageNet

Flatten Layer: הפלט של שכבת ' block5_pool' משוטחת לוקטור בודד.

Fully Connected Layers שלוש שכבות צפופות עם 4096 יחידות כל אחת, ואחריהן הפעלות של**Dropout layers, ReLU** לסניעת התאסת יתר.

Output Layer: שכבה צפופה סופית עם הפעלת softmax המחלקות, התואמת לסטפר המחלקות.(2)

שכבות הבסיס של VGG(5סקפואות (לא ניתנות לאיסון), וססקדות את האיסון בשכבות החדשות שנוספו הסחוברות בסלואן. הסודל סורכ<mark>ב</mark> עם אובדן קרוסנטרופיה קטגורי ו- Adam Optimizer .

האם אתן נעזרות בסודל קיים? איזה? סדוע בחרתן בו?

כן. VGG16 , הסודל היה יעיל יותר ססודל אחר שבנינו "סאפס" בגלל סשקלים סאוסנים סראש: VGG16 סגיע עם סשקולות שהוכשרו סראש ImageNet, שהוכשרו סראש בסערך הנתונים של ImageNet, הסכיל סיליוני תסונות באלפי קטגוריות. זה סספק נקודת התחלה חזקה לזיהוי תסונות.

קוד לבניית המודל:

```
myModel = Model(inputs=vgg16 base.input, outputs=predictions)
   return myModel
def compile model():
    model = build vgg16()
    model.compile(
       loss='categorical_crossentropy',
       optimizer='adam',
       metrics=['accuracy']
   model.summary()
   return model
def fit model():
    step size train = len(train batches)
    step size valid = len(test batches)
    model = compile model()
   history = model.fit(train_batches,
                       epochs=20,
                       steps per epoch=step size train,
                       validation data=test batches,
                       validation_steps=step_size_valid,
                       verbose=1)
    step size test = len(test batches)
   result = model.evaluate(test batches, steps=step size test)
    save_dir = os.path.join(os.getcwd(), 'build the model/build vgg16/saved_models')
   model name = 'vgg16-middlePart 200 images.h5'
   model path = os.path.join(save dir, model name)
   model.save(model_path)
   plt.figure(figsize=(10, 10))
   plt.plot(history.history['accuracy'], 'bo-', label='Train Accuracy - middle part vgg16')
   plt.plot(history.history['val_accuracy'], 'ro-', label='Test Accuracy - middle part vgg16')
   plt.xticks(range(0, 31))
   plt.xlabel('Number of epochs')
   plt.ylabel('Accuracy')
   plt.savefig("history middlepart 16 2 3")
```

```
batch size = 32 #32 של מונים של 52# batch size
image size = (224, 224)
classes = ['yes','no']
train_batches = ImageDataGenerator().flow_from_directory(
   directory='build the model/build vgg16/pics/train1',
   classes=classes,
   class mode='categorical',
   target_size=image_size,
   batch_size=batch_size,
   shuffle=True
test batches = ImageDataGenerator().flow from directory(
   directory='build the model/build vgg16/pics/test1',
   classes=classes,
   class mode='categorical',
   target_size=image_size,
   batch_size=batch_size,
   shuffle=False
def build vgg16():
מוגדר לתמונות בגודל 224*224 על 3 ערוצי צבע#
   vgg16 base = VGG16(input shape=(224, 224, 3), include top=True, weights='imagenet')
   vgg16_base = Model(inputs=vgg16_base.input, outputs=vgg16_base.get_layer('block5 pool').output)
   for layer in vgg16 base.layers:
       layer.trainable = False
   x = Flatten()(vgg16 base.output)
   x = Dense(4096, activation='relu')(x)
   x = Dropout(0.25)(x)
   x = Dense(4096, activation='relu')(x)
   x = Dropout(0.25)(x)
   x = Dense(4096, activation='relu')(x)
   x = Dropout(0.25)(x)
   predictions = Dense(len(classes), activation='softmax')(x)
```

עוד קצת קוז

פירוע...

המודל מבצע 20 טבבים (epchs).

ע"י: overfitting ע"י:

-Early_stopping: באם אין התקדמות במשך 3 סיבובים,

המודל מפסיק את הסבב.

-Dropout: הטתרת שכבות רנדומלית,

כדי לא להסתמך על נוירונים בודדים.

```
print("Test set classification accuracy: {0:.2%}".format(result[1]))
    return history, model

if __name__ == "__main__":
    history, model = fit_model()
    predictions = model.predict(test_batches, steps=len(test_batches), verbose=1)
    predicted_classes = numpy.argmax(predictions, axis=1)
    true_classes = test_batches.classes

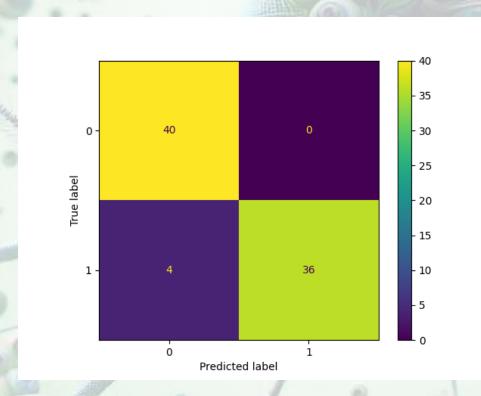
# confusion matrix
    confusion_matrix = confusion_matrix(true_classes, predicted_classes)
    cm_display = ConfusionMatrixDisplay(confusion_matrix=confusion_matrix)
    cm_display.plot()
    plt.savefig("cm_middlepart_16_200_images")
    plt.show()
```

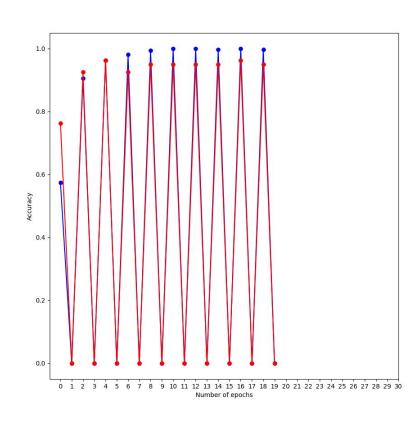
תוצאות המודל:

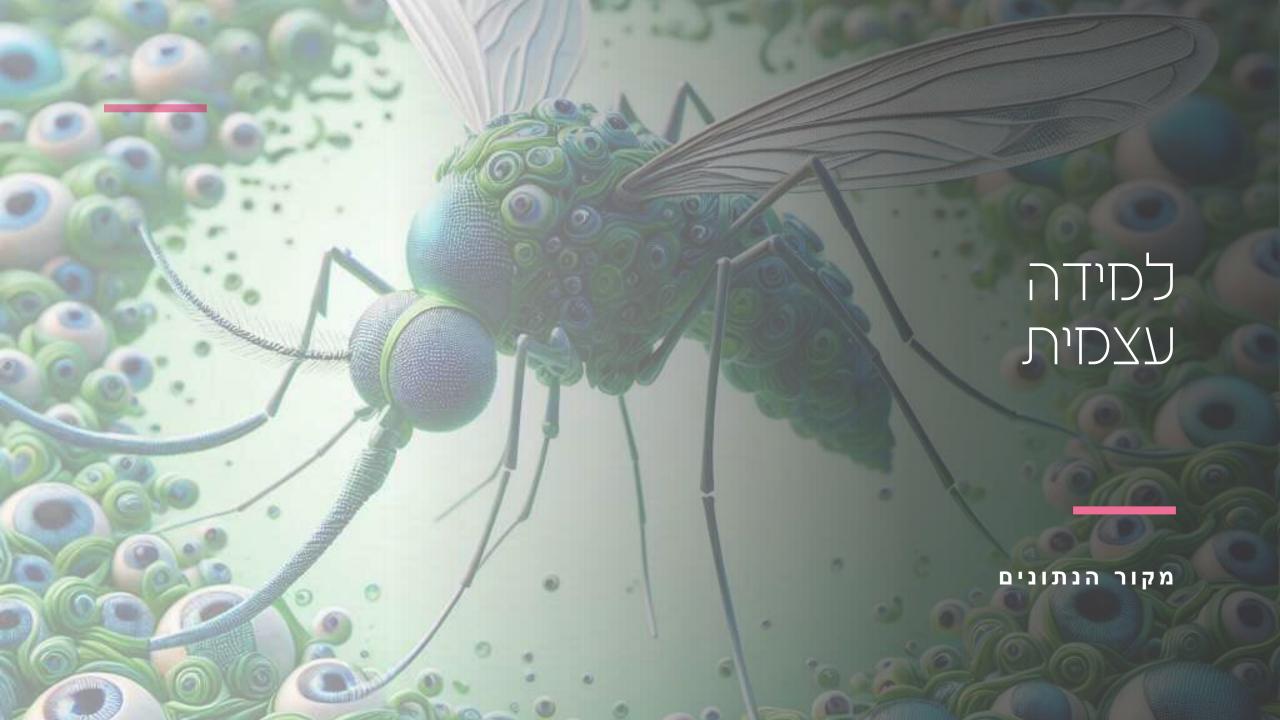
%פר הצלחה

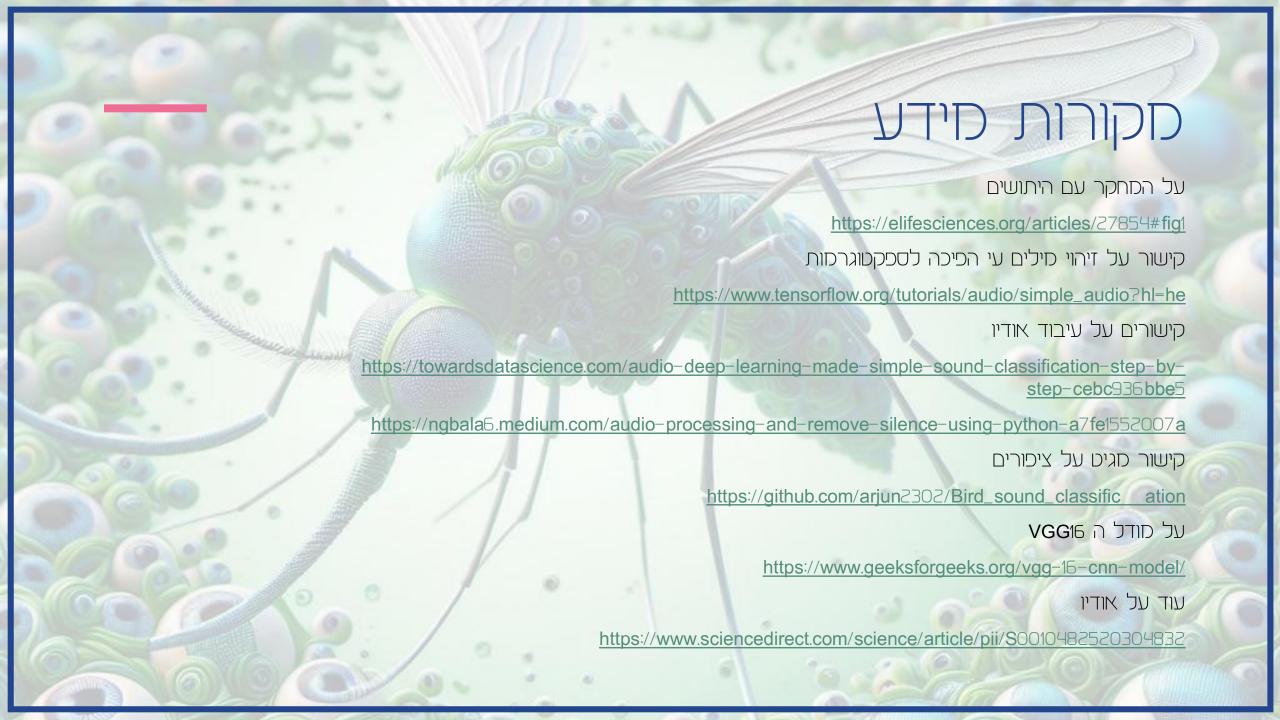
גרף להצגת סדדי הדיוק

לועריצת בלבול בבדיקת ה test









עיון בסקורות סידע/ סקור תיאורטי

סחקר סציג שיסוש בטלפונים ניידים לזיהוי וסיפוי סיני יתושים באסצעות הקלטות אקוסטיות של זסזום כנפיהם. טלפונים טלולריים, כולל דגסים פשוטים וזולים, סוצלחים קליט את צלילי היתושים באופן רגיש וספק סידע על זסן וסיקום הספגש. הסחקר סדגים זיהוי סינים בזסן אסת על בסיס הקלטות ונתונים סרחביים-זסניים שנאספים על ידי סשתסשים לא סקצועיים. הניסוי בשטח הוכיח שהשיטה יעילה וסהירה יותר סשיטות פיזיות. הפתרון סנצל את התשתית הגלובלית הקייסת של רשתות ניידות כדי לאפשר איסוף נתונים וסצויים באזורי סצוקה.

סיקרוסונים של טלפונים ניידים יכולים ללכוד בסדויק את צלילי פעיסות הכנפיים של יתושים, תוך ניצול רגישות הסכשיר בתוך רצועת תדר הקול שבה נופלים קולות היתושים. שיטה זו, בסיוע יישוסי הקלטה, סשתסשת בטרנספורסציה של פורייה לזסן קצר כדי לזהות תדרים ווריאציות אופייניים, סה שסקל על זיהוי סינים. בנוסף, טלפונים ניידים סתעדים אוטוסטית סטא נתונים רלוונטיים כסו זסן וסיקום, סה שסשפר את התועלת של הנתונים לסיפוי סרחבי-זסני של אוכלוסיות יתושים.



