



ספר פרויקט גמר - זיהוי חיות לפי הקלטות קוליות

המחלקה להנדסת חשמל ואלקטרוניקה

שם: אור כהן תעודת זהות: 318300373 חתימה: _____

בהנחיית: ד"ר יבגני גרשיקוב . חתימה: _____

כ"ז באלול תש"פ 16/09/2020

הוגש לשם מילוי חלקי של הדרישות לקבלת התואר
"בוגר במדעים B.Sc. בהנדסת חשמל ואלקטרוניקה"

תקציר:

הפרויקט בוצע במסגרת תכן הנדסי פנימי במכללת אורט בראודה כרמיאל. פרויקט זה עוסק בתכנון ופיתוח אפליקציה שמתמחה בזיהוי חיות לפי הקלטות קוליות שלהן. לעיתים קורה שאנחנו שומעים קולות של חיות, אבל איננו יודעים באיזו חיה מדובר, מה שיכול לעורר בנו תחושות של דאגה, פחד או חרדה. יש מקרים, במקומות ובזמנים מסויימים, שיש צורך לדעת באיזו חיה מדובר. לעיתים זה נובע מסקרנות ובמקרים אחרים, החיה יכולה להוות סכנה ממשית לבריאותנו. בכל המצבים האלה רצוי לזהות לפי קולות באלו חיות מדובר ולהגיב בצורה המתאימה במידת הצורך. הפרויקט הוא מערכת ממוחשבת, שבהינתן הקלטה של קולות של חיה, תזהה באיזו חיה מדובר ותספק מידע תמציתי עליה ועל צורת ההתנהגות המומלצת בקרבתה. למערכת תהיה יכולת למידה של קולות נוספים והוספתם למאגר הנתונים שלה.

תודות

ברצוני להודות לד"ר יבגני גרשיקוב, מנחה הפרויקט, אשר ליווה ותמך לאורך כל הדרך עם ייעוץ, הכוונה ועזרה.

תוכן

2.....	רשימת קיצורים:	
2.....	רשימת טבלאות:	
2.....	רשימת איורים:	
1.....	מבוא:	1.
2.....	רקע תיאורטי – מאפייני אותות קול:	2.
3.....	תיאור המערכת:	3.
3.....	הגדרה פונקציונאלית:	3.1.
3.....	מפרט פונקציונלי:	3.2.
3.....	מפרט טכני – פירוט מכלולי המערכת:	3.3.
4.....	דרישות המערכת:	3.4.
5.....	תרשים מלבנים של כלל המערכת:	3.5.
6.....	עיקרון הפעולה:	3.6.
6.....	הוראות הפעלה של המערכת:	3.7.
7.....	מטלות הנדסיות:	4.
7.....	מטלות הנדסיות ברמת מפרט הדרישות:	4.1.
7.....	ביצוע המטלות ע"י המתמחה:	4.2.
7.....	שלבי תכנון:	4.2.1.
10.....	תכן מפורט:	4.2.2.
16.....	בדיקות וניתוח המערכת:	5.
18.....	בעיות הנדסיות ופתרונותם:	6.
19.....	סיכום ומסקנות:	7.
19.....	אתגרים הנדסיים:	7.1.
19.....	הליך קבלת החלטות לאורך הפרויקט:	7.2.
20.....	סיכום:	7.3.
21.....	עמידה בדרישות:	7.4.
21.....	הצעות לשיפור:	7.5.
21.....	תרומת הפרויקט:	7.6.
21.....	מסקנות:	7.7.
23.....	סימוכין:	8.
24.....	נספחים:	9.

רשימת קיצורים:

MFCC – Mel Frequency Cepstrum Coefficients

STFT – Short Time Fourier Transform

GUI – Graphical User Interface

PC – Personal Computer

רשימת טבלאות:

- טבלה 1 מפרט טכני 3
טבלה 2 עמידה בדרישות 21

רשימת תרשימים:

- תרשים 1 תרשים מלבנים 5

רשימת איורים:

- איור 1 אחוזי דיוק/הפסדי המודל לסטים האימון/בדיקה כנגד כמות ה"אפוקים" 16
איור 2 תוצאת ניתוח המודל 16
איור 3 מטריצת הדיוקים לסיווג נכון פר חיה 17

1. מבוא:

פרויקט זה עוסק בתכנון ופיתוח אפליקציה שמתמחה בזיהוי חיות לפי הקלטות קוליות שלהן. לעיתים קורה שאנחנו שומעים קולות של חיות, אבל איננו יודעים באיזו חיה מדובר, מה שיכול לעורר בנו תחושות של דאגה, פחד או חרדה. יש מקרים, במקומות ובזמנים מסויימים, שיש צורך לדעת באיזו חיה מדובר. לעיתים זה נובע מסקרנות ובמקרים אחרים, החיה יכולה להוות סכנה ממשית לבריאותנו. בכל המצבים האלה רצוי לזהות לפי קולות באלו חיות מדובר ולהגיב בצורה המתאימה במידת הצורך. הפרויקט הוא מערכת ממוחשבת, שבהינתן הקלטה של קולות של חיה, תזהה באיזו חיה מדובר ותספק מידע תמציתי עליה ועל צורת ההתנהגות המומלצת בקרבתה. למערכת תהיה יכולת למידה של קולות נוספים והוספתם למאגר הנתונים שלה.

2. רקע תיאורטי – מאפייני אותות קול:

הדרך לתאר ולסווג אותות קול שונים, היא על פי מאפיינים (רבים) של אותות קול. המאפיינים מחולצים מן האות הן בתחום הזמן, הן בתחום התדר, והן בתחום הזמן והתדר (יחד). כאשר אנחנו רוצים לסווג חיות לפי הקלטות קוליות שהחיה משמיעה, עלינו להכניס בכניסת מכונת הלמידה העמוקה את המאפיינים של אות הקול ועל פי הם ללמד את המכונה לקשר בין המאפיינים (שבכניסת המערכת) לבין סוג החיה (יציאת המערכת). המאפיינים הבולטים שבהם, כמו כן אלו המאפיינים עליהם התבססתי בלמידת המכונה העמוקה הם: MFCC, CHROMA_STFT, MELSPECTOGRAM, SPECTRAL CONTRAST.

3. תיאור המערכת:

3.1 הגדרה פונקציונאלית:

המערכת תקלוט בזמן אמת את הקול אותו משמיעה החיה דרך המיקרופון, תבדוק את ההתאמה של המאפיינים (features) שלו למסד הנתונים שהוכן מראש תחזה את סוג החיה, ותספק מידע תמציתי עליה ועל צורת התנהגות מומלצת בקרבתה. המערכת תאפשר יכולת למידה של קולות נוספים והוספתם למאגר הנתונים שלה.

3.2 מפרט פונקציונלי:

- ✓ אפליקציה המתמחה בזיהוי חיות לפי הקלטות קוליות שלהן.
- ✓ ממשק גרפי נוח (GUI) להפעלת המערכת.
- ✓ יכולת קליטת קטע האודיו מהמיקרופון למחשב באמצעות ממשק גרפי למשתמש.
- ✓ יכולת לסנן רעשים לקבלת אות נקי של הקטע הנשמע.
- ✓ יכולת לזהות ולהבחין בין אירועי קול (קולות של חיה) לבין שקט ולחלץ את האירוע מהקטע הנשמע.
- ✓ יכולת הוצאת features (מאפיינים) של קטעי הקול השונים.
- ✓ יכולת השוואה של מאפייני קטעי הקול מול מאגר קיים.
- ✓ יכולת שמירת הקולות והמאפיינים שלהם למאגר.
- ✓ יכולת זיהוי של קולות החיות ב-75% דיוק או יותר עבור 5 חיות לפחות.
- ✓ יכולת לספק מידע תמציתי על החיה.
- ✓ יכולת לאמן מחדש את המערכת (בדאטא נוסף)

3.3 מפרט טכני – פירוט מכלולי המערכת:

סעיף	מכלולי המערכת	תפקיד
1	מיקרופון	קליטת קטע הקול בחלקו למחשב. ניתן להשתמש במיקרופון המובנה במחשב נייד או מיקרופון פשוט המתחבר למחשב חיצוני.
2	PC	משמש לכתובת התוכנית והפעלתה. נעבוד בסביבת Windows 10 עם דרייבר למיקרופון.
3	אלגוריתמים ב python. נשתמש באלגוריתמים קיימים של python או נכתוב חדשים.	שימוש במיקרופון לקבלת המידע, סינון רעשים, הוצאת features (מאפיינים) של קטע הקול, זיהוי וסיווג על סמך מאפיינים של קטעי הקול ועבודה עם מסד נתונים, החלטה לגבי שם החיה והצגת מידע תמציתי עליה. נעבוד בסביבת פיתוח python.

טבלה 1 מפרט טכני

3.4. דרישות המערכת:

- PC with Windows 10 Operation System
- משאבי עיבוד גבוהים למחשב
- סביבת עבודה python 3
- מיקרופון איכותי המתחבר למחשב לביצוע ההקלטות



3.6. עיקרון הפעולה:

פעולת המערכת מתבצעת ב3 שלבים .

בשלב הראשון – חילוץ ושמירת מאפיינים (features) מקבצי שמע במסד הנתונים והכנת מסד נתונים מעובד :

מסד הנתונים הגולמי עובר תהליך של חילוץ ושמירת features באמצעות מודולים שקיימים ברשות המערכת (מפורטים דרכי פעולתם בהרחבה בהמשך הספר). מסד הנתונים המעובד יהווה את סוף השלב הראשון.

השלב השני בניית/טעינת מודל מערכת למידה עמוקה מאומן :

השלב השני עוסק באלגוריתמים של למידה עמוקה על מנת ליצור/לטעון מודל מכונת למידה עמוקה מאומן שידע לחזות בהצלחה חיות לפי הקלטות קוליות שלהן. המודל המאומן יהווה את סוף השלב השני .

השלב השלישי – שימוש במערכת - חיזוי חיה מאות אודיו שנקלט ממיקרופון או שמור במחשב :

השלב השלישי הוא שימוש במערכת .

כאשר שני השלבים הקודמים בוצעו , נוכל להשתמש במודולים השונים שקיימים במערכת על מנת לסווג חיות לפי הקלטות קוליות שלהן , בשידור חי ממיקרופון או מקובץ השמור במערכת ואף לספק מידע תמציתי קצר על החיה.

3.7. הוראות הפעלה של המערכת:

על מנת להפעיל את המערכת נעבוד על ממשק המשתמש הגרפי GUI. **נעבוד בשני שלבים.**

שלב א' – יצירת מודל למידה עמוקה מאומן, מחולק ל2 אפשרויות:

-אפשרות א' , אימון מודל למידה עמוקה מחדש :

1.נבחר מסד נתונים איתו אנחנו רוצים לאמן את המודל שלנו.

2. נבחר האם לשמור את המודל המאומן או שלא.

3.נאמן את המודל שלנו

4. נבחר האם לקבל ניתוח של המודל שלנו.

-אפשרות ב' , טעינת מודל למידה עמוקה מאומן :

1.נבחר מודל למידה עמוקה לטעינה.

2.נטען את המודל.

שלב ב' , חיזוי וסיווג קטע חיה לפי הקלטה קולית :

1.נבחר אחת משני אפשרויות :

-אפשרות א' – הקלטת קטע קול חדש באמצעות מיקרופון שמחובר למחשב . (הערה -

נבחר האם לשמור את קטע הקול הזה)

-אפשרות ב' -טעינת קטע קול מהמחשב.

2.נבקש מהמערכת לבצע סיווג לקטע הקול הטעון , ונקבל מידע תמציתי עליה .

(הערה – נוכל לבקש לתת פידבק לתוצאת החיזוי , על מנת בסופו של דבר לאפשר

אימונים נוספים למערכת ולהרחיב אותה .)

4. מטלות הנדסיות:

4.1. מטלות הנדסיות ברמת מפרט הדרישות:

- כתיבת הצעת פרויקט
- למידת סביבת העבודה של python.
- למידת סביבת העבודה של google colab.
- בחירת מיקרופון ובדיקתו על ידי מימוש מודול ממשק למיקרופון והקלטה דרכו.
- בניית מסד נתונים.
- חקירה ובדיקה של features רלוונטים לצורך סיווג אותות הקול של החיות.
- בדיקה, איתור ותיקון תקלות בכל אלגוריתם מהאלגוריתמים שצוינו.
- תכנון וכתיבת ממשק גרפי לפרויקט GUI + אינטגרציה של כלל המודולים לממשק.
- כתיבת ספר פרויקט + מצגת.

4.2. ביצוע המטלות ע"י המתמחה:

4.2.1. שלבי תכנון:

- **בחירת סביבת פיתוח**
בחירת סביבת פיתוח של Python, מתוך שיקול של העמקת ידע בשפה ונוחות השפה, במיוחד בנוגע לכל התחום של למידה עמוקה, בו השפה תומכת בצורה מיטבית. בחירת סביבת פיתוח בgoogle colab על מנת להריץ את קטעי הקוד על שרתים "רחוקים" שיכולים לעבד בצורה טובה ומהירה את כלל הפונקציות.
- **מציאת/בניית מסד נתונים בצורתו הגולמית (Dataset)**
מציאת מסד נתונים רלוונטי, גדול מספיק ואיכותי המכיל קולות של חיות.
- **בניית מודול לחילוץ features (מאפיינים) הרלוונטים מקטע אודיו**
המודול אחראי להוצאת features הרלוונטים מקובץ אודיו.
- **בניית מודול לחילוץ ושמירת features (מאפיינים) לכל קטעי האודיו המכילים חיות במסד הנתונים הגולמי.**
המודול אחראי להוצאת features הרלוונטים לקבצי האודיו שמכילים קולות של חיות מתוך מסד הנתונים הגולמי.
את features האלה הוא שומר בקבצים מעובדים לצורך שליפה מהירה שלהם, ובכך הוא יוצר "מסד נתונים מעובד" שממנו מכונת הלמידה העמוקה תלמד.
- **בניית מודול לטעינת "מסד הנתונים המעובד" שכולל בו features (מאפיינים) לכל קטעי האודיו המכילים חיות ממסד הנתונים**
מודול אחראי לטעינת "מסד הנתונים המעובד" שממנו מכונת הלמידה העמוקה תלמד.

- **בניית מודול שמפצל את "מסד הנתונים המעובד" לאימון/בדיקה של המכונה**
מודול זה אחראי לחלק את "מסד הנתונים המעובד" לנתונים שישמשו ללימוד מכונת הלמידה העמוקה, ולבדיקה של המכונה.
- **בניית מודול הבונה ומקמפל את מכונת הלמידה העמוקה**
מודול זה אחראי לבניית מכונת למידה עמוקה שתהיה מתאימה לבעיית הסיווג שמתוארת בפרויקט (מבחינת מודל המכונה, מספר וסוג השכבות, פונ' האקטיבציה של השכבות ועוד).
בנוסף מודול זה מקמפל את מודל המכונה.
- **בניית מודול הטוען מודל מכונת למידה עמוקה מאומן או לחילופין מאמן ושומר מכונת למידה**
מודול זה אחראי לטעון מודל מכונת למידה עמוקה מאומן שמור מראש, על מנת לקצר תהליכי עיבוד / לחסוך זמן יקר של לימוד המכונה.
בנוסף מודול זה אחראי (במקרי הצורך) לאמן מודל לא מאומן, ולשמור אותו.
במידה ונשמור את המודל, יוצר עוד מודול שיכיל קבצי מכונות למידה עמוקות מאומנות.
- **בניית מודול לניתוח מודל מכונת הלמידה העמוקה**
מודול זה אחראי לנתח את מודל מכונת הלמידה העמוקה המאומנת.
מודול זה מוציא נתונים לגבי אחוזי דיוק (לאימון ולבדיקה), אחוזי כישלון (לאימון ולבדיקה), מציג מטריצת חיזויים אל מול התוצאות הנכונות של החיזויים, על מנת למשל לנתח ולהבין בבירור אחוזי דיוק פר חיה.
- **מודול קליטת אות שמע ממיקרופון**
מודול זה אחראי לקלוט אות שמע ממיקרופון המחובר למחשב.
במודול זה אות השמע הזה מנוקה מרעשים, וביציאת המודול נקבל אות שמע המכיל אירוע (קול של חיה) בצורה מיטבית לקראת השלב הבא בו יחולצו מן אות זה המאפיינים שלו (features), אלה יעברו למכונת הלמידה העמוקה והיא תחזה את החיה לפיו.
- **מודול חיזוי סוג חיה לפי הקלטה קולית**
מודול זה אחראי לקבל אות שמע, להוציא ממנו את features הרלוונטים (באמצעות המודול המתאים), להכניס בכניסת מכונת הלמידה העמוקה, וביציאתה להחליט על איזו חיה מדובר.
- **מודול ממשק משתמש גרפי**
מודול זה אחראי לחבר את כל המודולים האחרים שבמערכת, לבצע את תהליך הבקרה והלוגיקה לקריאה למודול המתאים בתהליך ומהווה ממש פשוט למשתמש לתפעול המערכת ולקבלת התוצאות המתאימות.
- **מודולים נוספים שנוצרו במהלך פיתוח/הרצת המערכת (עליהם אפרט בהמשך):**
 - **מודול שמירת אות אודיו לקובץ מסוג wav על המחשב**
 - **מודול "מסד הנתונים המעובד" שכולל בו features (מאפיינים) לכל קטעי האודיו המכילים חיות ממסד הנתונים המקורי.**

מודול זה נוצר כאשר נבחר באפשרות של שמירת מסד הנתונים המעובד ב"מודול לחילוץ ושמירת features (מאפיינים) לכל קטעי האודיו המכילים חיות במסד הנתונים הגולמי."

- **מודול שמכיל מודלים למכונת למידה עמוקה מאומנים.**
מודול זה נוצר כאשר נבחר באפשרות של שמירת מודל של מכונת למידה עמוקה, במודול "הטוען מודל מכונת למידה עמוקה מאומן או לחילופין מאמן ושומר מכונת למידה לא מאומנת"
- **מודול שמכיל הקלטות קול שנקלטו מהמיקרופון ונשמרו על ידי " מודול עזר –**
מודול שמירת אות אודיו לקובץ מסוג wav על המחשב"
- **מודול המכיל מידע תמציתי קצר על כל סוגי החיות.**

4.2.2. תכן מפורט:

- **בניית מסד נתונים בצורתו הגולמית - "Dataset"**

זיהוי אירועי שמע וסיווג צלילים הם משימה קשה ללמידת מכונה .
אחד ההיבטים המאתגרים ביותר הוא למצוא מסד נתונים שהוא מספיק טוב(מבחינה איכותית וכמותית) ומתאים לעבודה עם התוויות שברצוני לסווג .
לאחר מאמצים ונסיונות רבים למציאת מסד נתונים מתאים (יש המון מסדי נתונים שנתקלתי בהם ברחבי האינטרנט וביוטיוב , רובם היו לא מתאימים או לא טובים מבחינה איכותית וכמותית), בחרתי להתמקד ולעבוד על מסד הנתונים של "Kaggle Environmental Sound Classification 50".
מסד נתונים זה ("Kaggle Environmental Sound Classification 50") – כולל 2,500 קבצי wav , שנדגמו בתדר 16khz עבור 50 תוויות שונות ומגוונות (חלקם מכילים בעלי חיים אך רובם קולות שאינן בעלי חיים).
מסד הנתונים מאוזן , כל תווית מכילה 40 קבצי אודיו , בעלי 50 שניות , או במילים אחרות לכל תווית יש אודיו כולל בזמן של 3 דקות ו20 שניות.
לכן מבחינת איזון – עם מסד נתונים זה טוב לעבוד.
מסד נתונים זה כולל 12 סוגי החיות : עורב , פרה , חתול, ציפורי שיר , חזיר , צרצר , צפרדע , חרקים , תרנגול , תרנגולת , כלב , כבשה .
לכן מבחינת מגוון החיות – מסד נתונים זה מתאים לצרכי בפרויקט .
בנוסף (מעבר לקולות של חיות) מסד נתונים זה מכיל קולות שאינם חיות עליהם אני מדלג ולא מוציא מהם מאפיינים.
- **בניית מודול לחילוף features (מאפיינים) הרלוונטים מקטע אודיו. "extract_features"**

המודול אחראי להוצאת הfeatures הרלוונטים מקובץ אודיו .
(כניסה – קובץ אודיו , יציאה – features)
כפי שצינתי במודול הקודם , האתגר הגדול בלבנות תוכנת למידה עמוקה לסיווג קולות הוא מציאת מסד נתונים מתאים לעבודה מבחינת : רעש , ואיזון הנתונים שבו.
לכן לפני הוצאת הfeatures (מאפיינים) המתאימים קודם אנו צריכים לטפל ב"בעיית הרעש" שעלולה להיות באות השמע .
על מנת לנקות רעשים לא רצויים מאות השמע (כמובן שצריך לבצע את אותו אלגוריתם על כל קובץ ממסד הנתונים בנפרד) תחילה נטען את קובץ הwav באמצעות "Librosa" (ספרייה מאוד נוחה ושימושית בפייתון שמתעסקת בתחומי עיבוד אותות שמע), ונמיר את אות השמע מאיך שהוא מוצג בקובץ הwav למערכי "numpy" (שזוהי ספרייה מאוד נוחה ושימושית בפייתון להציג מערכים ווקטורים ולבצע פעולות רבות איתם) שמכילים את האמפליטודה של האות שנדגם על פי תדר הדגימה הרצוי (אצלי בפרויקט 22,050Hz).
לאחר שיש מערכי "numpy" מוכנים אשר מכילים את המידע הנדרש מן האות בתחום הזמן נשתמש באלגוריתם של ניקוי רעשים שנקרא "Spectral Noise Gating" שקיים בספרייה של "Audacity".
אלגוריתם זה מבצע – תחילה דגימה של רעשי רקע מאות השמע - זאת תהיה חתימת הרעש שהאלגוריתם ישתמש בה על מנת לדכא רעשים מאותה חתימה לאורך כל האות . לפי

ספקטרום התדרים של דגימה זו, האלגוריתם יודע לדכא בצורה של סגמנטים באות השמע את המרכיבים הרועשים שבאות ולהנחית אותם.

על מנת לבצע את האלגוריתם הזה נשתמש בספרייה "Noisereduce" שבה יש מימוש לאלגוריתם המדובר.

ניקוי הרעשים הוא תהליך ארוך ומייגע, פרטני לקובץ אודיו אחד בכל פעם, והוא לא יעיל למסד נתונים גדול, לכן אבחר שלא להשתמש בו במהלך עיבוד מקדים של מסד הנתונים. במודולים הבאים אציג איך כן בכל זאת אשתמש בעיבוד המקדים שתואר כאן (מודול "קליטת אות שמע ממיקרופון").

לאחר שאות השמע נקי מרעש, נרצה להוציא מן האות מאפיינים (features) בתחום הזמן והתדר.

כאשר אנחנו רוצים לסווג חיות לפי הקלטות קוליות שהחיה משמיעה, עלינו להכניס בכניסת מכונת הלמידה העמוקה את המאפיינים של אות הקול ועל פי הם ללמד את המכונה לקשר בין המאפיינים (שבכניסת המערכת) לבין סוג החיה (יציאת המערכת). המאפיינים הבולטים שבהם, כמו כן אלו המאפיינים עליהם התבססתי בלמידת המכונה העמוקה הם:

- MFCC
- CHROMA_STFT
- MELSPECTOGRAM
- SPECTRAL CONTRAST

כולם בוצעו בעזרת "הסתכלות" על חלון FFT בגודל $\text{fft} = 2048$ דגימות, קפיצה בין חלונות של $\text{hop_length} = 512$ דגימות ותדר דגימה $\text{sr} = 22,500\text{Hz}$.

ביציאת המודול, בצורה של מערך, מחזיר את 187 הfeatures שחילץ מקובץ האודיו.

- **בניית מודול לחילוץ ושמירת features (מאפיינים) לכל קטעי האודיו המכילים חיות במסד הנתונים הגולמי.**

מודול זה מכין מסד נתונים מעובד, ממסד הנתונים הגולמי, כאשר הוא משתמש ונעזר במודול הקודם שתפקידו לעבד ולחלץ מאפיינים (features) מאות שמע בודד. את מסד הנתונים המעובד הנל (אם נרצה, וכמובן שנעדיף את האפשרות הזו) נוכל לשמור בקבצים לצורך שליפה מהירה שלהם בהמשך לצורך אימון מכונת הלמידה העמוקה.

תחילה אסביר על מסד הנתונים הגולמי שלי.

מסד הנתונים שבחרתי כולל בו בנוסף ל12 סוגי החיות שאותם ארצה לסווג, 38 סוגים של קולות שאינן חיות.

לכן, לא ארצה לעבוד עם כל מסד הנתונים בצורתו הגולמית, אלא רק עם קבצי הקול שמכילים חיות.

מסד הנתונים, כפי שהורד מאתר Kaggle כולל בתוכו בנוסף למאגר קבצי האודיו, קובץ csv שמכיל מידע עבור כל קובץ אודיו - האם הוא קובץ אודיו שמכיל קול של חיה, ואם כן על

איזו חיה מדובר .

לכן בעזרת מודול "prepare_dataset", אעבור ואחלץ *רק מקטעי האודיו שמכילים חיות* את features למכונה.

קטע הקוד שבמודול זה עובר בתיקיית מסד הנתונים, ועבור כל קובץ לפני שהוא שואב ממנו את המידע הנדרש (features), הוא בודק בקובץ csv האם הוא קטע אודיו שמכיל חיה מסוימת, אם כן הוא מחלץ את features ממנו.

בסופו של האלגוריתם, יש לנו 4 פרמטרים מוחזרים שאנחנו צריכים בהמשך והם: "np_arr_all_data", "np_arr_all_labels", "np_arr_all_numeric_labels", "classes" הפרמטר הראשון – הוא מערך של numpy, והוא מכיל מערכים של features של כל קטע אודיו שמכיל קול של חיה.

הפרמטר השני, בקורלציה של אינדקסים לפרמטר הראשון, מכיל את התווית של קטע האודיו. הפרמטר השלישי, שוב בקורלציה לפרמטר הראשון והשני, מכיל את המספר (0-11) שמתייג את התווית של כל חיה.

הפרמטר הרביעי – הוא מערך של שמות של החיות – והוא משמש כמקודד בין הפרמטר השלישי (מספר נומרי) לבין סוג החיה שהמספר מסמל.

שלושת הפרמטרים הראשונים, נשמרים בקבצי numpy בתיקייה שנקראת "prepared_dataset".

לאחר שסיימנו עם התכנית הזאת, יש לנו מסד נתונים מעובד ומוכן למכונת הלמידה העמוקה.

את מסד הנתונים המעובד הנל (אם נרצה, וכמובן שנעדיף את האפשרות הזו) נוכל לשמור בקבצים לצורך שליפה מהירה שלהם בהמשך לצורך אימון מכונת הלמידה העמוקה.

- **בניית מודול לטעינת "מסד הנתונים המעובד" שכולל בו features (מאפיינים) לכל קטעי האודיו המכילים חיות ממסד הנתונים**

במקביל למודול הקודם, יש במערכת גם מודול שנקרא "load_prepared_dataset" – ותפקידו הוא לאחר שמשתמשים במודול הקודם ומכניסים מסד נתונים מעובד שמוכן לכניסה למודל של מערכת למידה עמוקה, נוכל לשאוב את הדאטאסט המעובד, מבלי לעבד אותו שוב פעם (כי זהו תהליך ארוך ותוצאתו קבועה).

מודול זה אחראי לחסוך זמן ועלות עיבוד גבוהה ויקרה. המודול טוען קבצי npy למערכי numpy שיהיו כניסה למכונת הלמידה העמוקה. כמו כן הוא מחשב מחדש את הפרמטר "classes" שתפקידו פורט במודול הקודם.

- **בניית מודול שמפצל את "מסד הנתונים המעובד" לאימון/בדיקה של המכונה**

לאחר הכנת מסד הנתונים המעובד בעזרת המודולים שתוארו למעלה, נחלק אותו למסד נתונים לאימון, ומסד נתונים לבדיקה, לקראת ההכנסה של הנתונים למודל, באמצעות קטע הקוד שבמודול "divide_dataset".

החלטתי לחלק את הדאטאסט באחוזים של 80% לאימון, ו20% לבדיקה.

- **בניית מודול הבונה את מכונת הלמידה העמוקה**

מודול זה אחראי לבניית מכונת למידה עמוקה שתהיה מתאימה לבעיית הסיווג שמתוארת בפרויקט (מבחינת מודל המכונה, מספר וסוג השכבות, פוני האקטיבציה של השכבות ועוד). בנוסף מודול זה מקמפל את מודל המכונה.

המודל שלי הוא מודל שכבות "רציף" שנבנה על ידי ספריה של "keras".

המודל כולל 4 שכבות "צפופות".

השכבה הראשונה מכילה 256 נוירונים, כאשר המימד של הכניסה לשכבה הראשונה נקבע דינאמית לפי כמות הfeatures (מאפיינים) שבחרתי לעבוד איתם (בפרויקט כמות הfeatures שווה 187).

השכבה השנייה מכילה 128 נוירונים.

השכבה השלישית מכילה 64 נוירונים.

השכבה הרביעית מכילה 12 נוירונים (כמספר הקטגוריות (סוגי החיות) בפרויקט שלי).

כל שכבה מכילה פונקציית אקטיבציה מסוג "Relu".

וכאשר השכבה האחרונה מכילה פונקציית אקטיבציה מסוג "softmax".

המודל מקומפל על ידי פונקציית הפסד "sparse_categorical_crossentropy", עם

אופטימיזר של "Adam".

קצב הלמידה שבחרתי הוא $\text{learning_rate} = 0.001$ קבוע.

$\text{epochs} = 50$ (מצאתי כאופטימלי).

$\text{batch_size} = 60$ (מצאתי כאופטימלי).

בסוף התכנית מוחזר המודל בנוי ומקומפל, ומוכן לאימון.

- **בניית מודול הטוען מודל מכונת למידה עמוקה מאומן או לחילופין מאמן ושומר מכונת למידה לא מאומנת**

מודול זה אחראי לטעון מודל מכונת למידה עמוקה מאומן ושומר מראש, על מנת לקצר תהליכי עיבוד / לחסוך זמן יקר של לימוד המכונה.

בנוסף מודול זה אחראי (במקרי הצורך) לאמן מודל לא מאומן, ולשמור אותו במודול נפרד של מודלים למכונות למידה מאומנות.

*קובץ מודל נשמר בפורמט hdf5.

כל היכולות הללו ממומשות בקטע הקוד "create_or_load_fitted_model" שבמודול.

- **בניית מודול לניתוח מודל מכונת הלמידה העמוקה**

לתפקיד המתואר לניתוח מודל מכונת הלמידה העמוקה: קיימים 2 סאב-מודולים.

אחד בשם "model_evaluation" ואחד בשם "analyze_model".

מודולים אלה אחראים לנתח את מודל מכונת הלמידה העמוקה המאומנת.

מודולים אלה מוציאים נתונים לגבי אחוזי דיוק (לאימון ולבדיקה), אחוזי כישלון (לאימון ולבדיקה), מציגים מטריצת חיזויים אל מול התוצאות הנכונות של סיווג החיה, ובנוסף

מציגים את "ההיסטוריה" של המודל ממנה ניתן ללמוד המון.

כל זאת על מנת לנתח ולהבין בבירור אחוזי דיוק פר חיה ולהסיק מסקנות לגבי המודל.

במודול "model_evaluation" – מוצגים אחוזי דיוק וכישלון לסט האימון ולסט הבדיקה .
במודול "analyze_model" – זהו מודול שמאחד 2 פונקציות שאחת מהן "plot_history"
שתפקידה להציג את "ההיסטוריה של המודל" – דיוקים והפסדים במהלך האימון והבדיקה.
והשנייה "plot_confusion_matrix" שתפקידה להציג את מטריצה הדיוקים פר חיה לאחר
אימון .

- **מודול קליטת אות שמע ממיקרופון**

מודול זה אחראי לקלוט אות שמע ממיקרופון המחובר למחשב .
במודול זה אות השמע הזה מנוקה מרעשים , וביציאת המודול נקבל אות שמע המכיל אירוע
(קול של חיה) בצורה מיטבית לקראת השלב הבא בו יחולצו מן אות זה המאפיינים שלו
(features), אלה יעברו למכונת הלמידה העמוקה והיא תחזה את החיה לפיו.

ניקוי הרעשים - על מנת לדכא רעשי רקע לקראת הכנסת קטע הקול שארצה לסווג למכונת
הלמידה העמוקה אשתמש באלגוריתם המפורט במודול " בניית מודול לחילוץ features
(מאפיינים) הרלוונטים מקטע אודיו "extract_features".

במודול זה ישנו עוד תהליך נוסף שאעשה והוא חיתוך קטע האודיו המוקלט לחלקים שבהם
נמצאים "אירועי קול", בהם אבחין על ידי : ניקוי רעשים , חיתוך וטיפול האודיו על פי
הרעשים שבין האירועים (בהנחה שבין אירוע לאירוע קיים שקט)
כל זאת על מנת שמכונת הלמידה העמוקה , במקרה שיש אות שמע שכולל בו הרבה "שקט"
(לאחר הפחת הרעשים) , לא תזהה לי את השקט הזה כסוג של חיה אלא רק את "אירוע
הקול".

את אות האודיו הנל (לאחר ניקוי רעשים וחיתוך) , נעביר במודול חילוץ המאפיינים
(features) , ומשם למודול "חיזוי סוג החיה לפי הקלטה קולית".
הערה – יכול להיות בין שני המודולים מודול ביניים "מודול שמירת אות אודיו לקובץ מסוג
wav על המחשב" , זאת מכיוון בעיה הנדסית שחוייתי (מפורטת ב"בעיות ההנדסיות ופתרונן"
הקשורה ליכולות העיבוד הנמוכות של המחשב שלי והצורך בgoogle colab להרצת
המערכת.

- **מודול חיזוי סוג חיה לפי הקלטה קולית**

מודול זה אחראי לקבל אות שמע, להוציא ממנו את features הרלוונטים (באמצעות המודול
"extract_features" , להכניס בכניסת מכונת הלמידה העמוקה , וביציאתה להחליט על איזו
חיה מדובר.
מודול זה משתמש במתודת חיזוי (predict) של המודל של keras .

- **מודול ממשק משתמש גרפי**

מודול זה אחראי לחבר את כל המודולים האחרים שבמערכת , לבצע את תהליך הבקרה
והלוגיקה לקריאה למודול המתאים בתהליך ומהווה ממשק פשוט למשתמש לתפעול
המערכת ולקבלת התוצאות המתאימות.
(על מודול זה מפורט בהרחבה ב"הוראות ההפעלה" של המערכת)

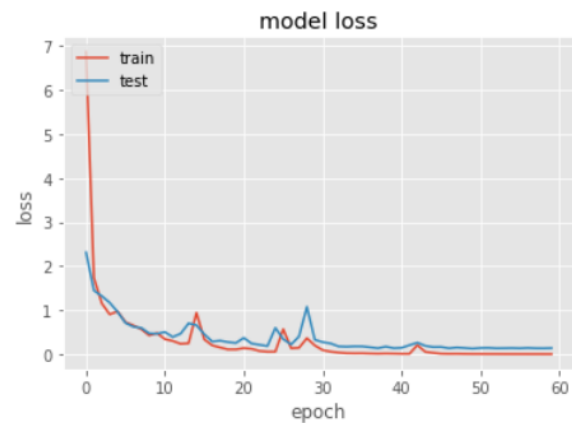
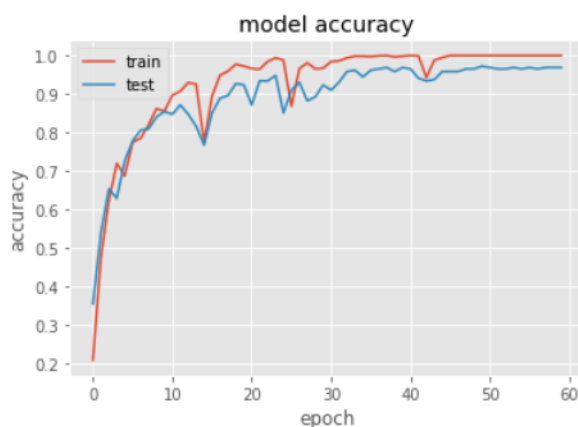
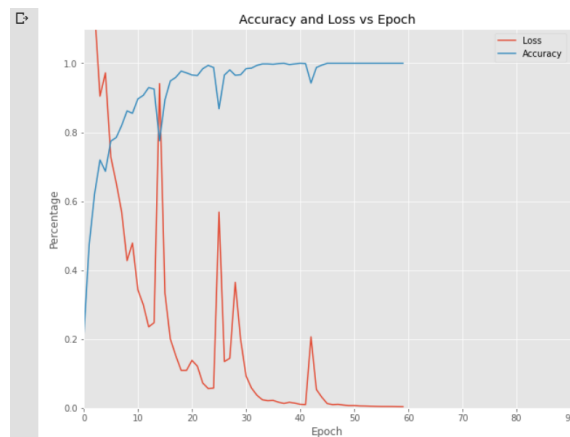
- ***מודול עזר – מודול שמירת אות אודיו לקובץ מסוג wav על המחשב**
 - עבודה עם google colab לא מאפשרת שימוש בחומרת המחשב כגון הפעלה של מיקרופון , לכן יצרתי מודול ביניים (בין מודול "קליטת אות שמע ממיקרופון" לבין מודול "חיזוי סוג חיה לפי הקלטה קולית").
 - מודול זה אחראי לשמירת אות אודיו לקובץ מסוג wav על המחשב , לכן הוא יוצר מודול חדש שמכיל הקלטות של קולות שהתבצעו מהמיקרופון.
 - כך בשני חלקים אוכל להפעיל את המערכת שלי .
 - החלק הראשון פועל על המחשב האישי – הקלטת קול דרך המיקרופון ושמירתו לקובץ wav על המחשב .
 - החלק השני פועל בסביבת העבודה של google colab ומבצע חיזוי על אות הקול שהוקלט ונשמר במחשב.
 - לחילופין – ניתן לא להשתמש במודול זה ולהריץ את המערכת על מחשב בעל יכולות עיבוד גבוהות.
- **ישנם עוד כמה מודולים שנוצרים במהלך הרצת התוכנית והם :**
 - **מודול "מסד הנתונים המעובד" שכולל בו features (מאפיינים) לכל קטעי האודיו המכילים חיות ממסד הנתונים המקורי.**

מודול זה נוצר כאשר נבחר באפשרות של שמירת מסד הנתונים המעובד ב"מודול לחילוץ ושמירת features (מאפיינים) לכל קטעי האודיו המכילים חיות במסד הנתונים הגולמי."
 - **מודול שמכיל מודלים למכונת למידה עמוקה מאומנים.**

מודול זה נוצר כאשר נבחר באפשרות של שמירת מודל של מכונת למידה עמוקה , במודול "הטוען מודל מכונת למידה עמוקה מאומן או לחילופין מאמן ושומר מכונת למידה לא מאומנת"
 - **מודול שמכיל הקלטות קול שנקלטו מהמיקרופון ונשמרו על ידי " מודול עזר – מודול שמירת אות אודיו לקובץ מסוג wav על המחשב"**
 - **מודול המכיל מידע תמציתי קצר על כל סוגי החיות.**

5. בדיקות וניתוח המערכת:

המערכת המתוארת הנל, הצליחה לעמוד בדרישות הפרויקט, והשיגה את מטרותו - מודל למידה עמוקה מאומן שמצליח לחזות ולסווג חיות לפי הקלטות קוליות שלהן. המודל הצליח לעמוד בדרישת הדיוק בהצלחה – מעל 75% דיוק ל 5 חיות לפחות. מצורפים כאן איורים המנתחים את המערכת.



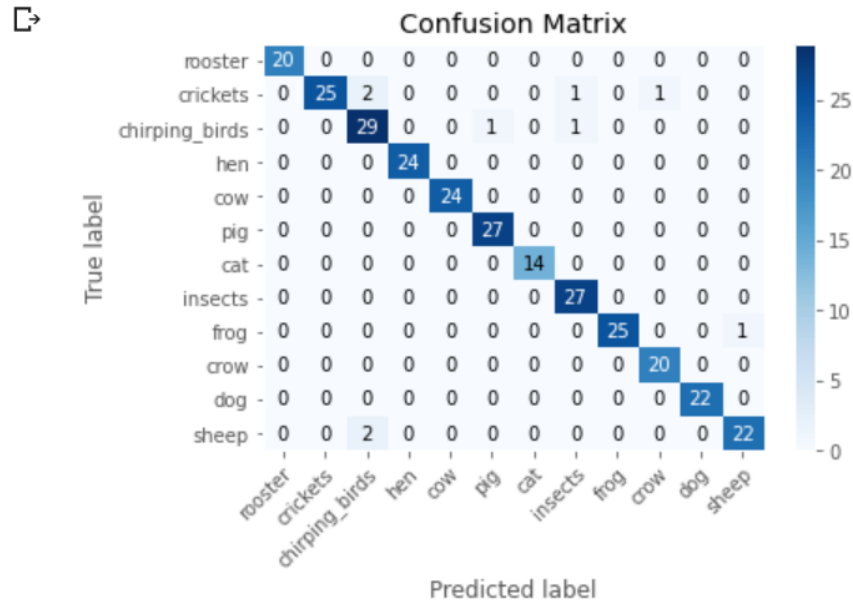
איור 1 אחוזי דיוק/הפסדי המודל לסטים האימון/בדיקה כנגד כמות ה"אפוקים"

➡ Training Accuracy: 1.0
Training Cross Entropy: 0.00
Testing Accuracy: 0.96875
Testing Cross Entropy: 0.14

איור 2 תוצאת ניתוח המודל

ניתן לראות כי אחוז הדיוק הכללי של המערכת (בשקלול כל סוגי החיות האפשריות) מגיע למעל 90%.

ניתן להראות גם את אחוזי הדיוק של המודל פר חיה בעזרת מטריצת הדיוקים ולהסיק מסקנות נוספות על חיות ספציפיות.



איור 3 מטריצת הדיוקים לסיווג נכון פר חיה

6. בעיות הנדסיות ופתרונותם:

1. עיבוד מסד נתונים גדול ומודלים של למידה עמוקה מצריכים יכולות עיבוד גבוהות מאוד .
הפתרון : פיתוח בסביבת עבודה של google colab , בה ניתן להריץ ולהשתמש במשאבי עיבוד הנמצאים בשרתים של החברה .
2. עבודה עם google colab לא תמיד מאפשרת לייבא ספריות וחבילות הקיימות בפייתון .
הפתרון : יבוא פתרונות/ספריות/חבילות חלופיים .
3. עבודה עם google colab לא מאפשרת שימוש בחומרת המחשב כגון הפעלה של מיקרופון ,
לכן לא הייתה לי אפשרות לעשות debugging עם המודול שקולט אות שמע מהמיקרופון
וכמובן להפעיל אותו, ולא לעשות debuggibg על כלל המערכת ממשק הGUI ולהפעיל את
כלל המערכת על google colab ובכך להיעזר במשאבי העיבוד שנדרשים .
פתרון : תכנון וביצוע מודול שמירת אות אודיו לקובץ מסוג wav על המחשב , כך שאוכל
לעשות debugging ולהריץ את המערכת בשני חלקים : החלק הראשון פועל על המחשב
האישי – הקלטת קול דרך המיקרופון ושמירתו לקובץ wav על המחשב .
החלק השני פועל בסביבת העבודה של google colab ומבצע חיזוי על אות הקול שהוקלט
ונשמר במחשב.
או לחילופין – להריץ את המערכת על מחשב בעל יכולות עיבוד גבוהות.

7. סיכום ומסקנות:

7.1. אתגרים הנדסיים :

1. עבודה עם סביבת העבודה של google colab .
2. עבודה בתחום עיבוד אותות השמע הבנת הרקע התיאורטי והטמעתו בפיתוח מערכת בסביבת עבודה של python .
3. עבודה והכרה עם סוגי מאפיינים (features) שונים של אותות שמע.
4. חקירת אלגוריתמים שונים להוצאת מאפיינים (features) הטובים לזיהוי החיה .
5. מציאת מסד נתונים גדול ואיכותי שמתאים למערכת מבוססת למידה עמוקה , לכן מסד הנתונים צריך להיות איכותי מבחינת רעש , ומאוזן מבחינת סוגי התוויות שעל המערכת לדעת לסווג.
6. עבודה עם מסד נתונים גדול.
7. עבודה עם מערכת בינה מלאכותית (למידה עמוקה) .
8. הכרת אלגוריתמי סיווג בלמידה עמוקה.
9. בחינת אלגוריתמים אלו לכדי הגעה ליעד- זיהוי ב- 75% הצלחה או יותר עבור 5 חיות לפחות.

7.2. הליך קבלת החלטות לאורך הפרויקט:

- בפרויקט זה התמודדתי עם בחירות והחלטות רבות לאורך כל הדרך :
- בחירת פרויקט המתמחה בעיבוד אותות ומודלים של למידה עמוקה.
 - בחירת סביבת עבודה של python .
 - בחירת סביבת עבודה בgoogle colab .
 - קביעת זמני עבודה עבור כל שלב וסדר פעולות הביצוע.
 - קביעת דרישות היכולות הפונקציונליות של הפרויקט .
 - בחירת מסדי נתונים שיתאימו לי לפרויקט בצורה הטובה ביותר.
 - בחירת סוגי מאפייני אותות השמע (feature) שיתאימו לי לפרויקט בצורה טובה.
 - בחירת מודלים ומאפיינים שלהם של למידה עמוקה ואלגוריתמי סיווג המתאימים ביותר לפרויקט.
 - בחירת עיצוב , ותפעול הממשק הגרפי למשתמש GUI שיתאימו בצורה הנוחה ביותר לצורך הפעלת המערכת.

כל שלב והחלטה התקבלה לאחר תהליך מחשבתי רב, ולאחר ניסויים רבים עד לקבלת התוצאה הרצויה.

7.3. סיכום:

פרויקט זה עוסק בפיתוח מערכת לזיהוי חיות על פי הקלטות קוליות שלהן. הפרויקט הוא מערכת ממוחשבת, שבהינתן הקלטה של קולות של חיה, תזהה באיזו חיה מדובר ותספק מידע תמציתי עליה ועל צורת ההתנהגות המומלצת בקרבתה. המערכת תקלוט בזמן אמת את הקול אותו משמיעה החיה דרך המיקרופון, תחלץ ממנו את המאפיינים (features) שלו, תבדוק את ההתאמה שלהם למסד הנתונים שהוכן מראש (מסד נתונים זה תחילה היה בצורתו הגולמית, אך לאחר עיבוד שלו יצרתי מסד נתונים מעובד ו"מוכן לעבודה") ובעזרת מודל של מכונת למידה עמוקה (שאותו כמובן צריך לאמן/לטעון מאומן לפני שימוש) תצליח לסווג את אות הקול לסוג של חיה ותספק מידע תמציתי עליה ועל צורת התנהגות מומלצת בקרבתה. המערכת תאפשר למידה של מאפייני הקוליים של האות הנקלט והוספה שלהם למאגר.

המערכת עומדת בכלל היכולות הפונקציונליות והן בדרישות שנקבעו בהצעת הפרויקט ואף מעבר לנדרש.

פרויקט זה עובד בסביבת עבודה של python, כאשר חלק מהתוכניות הורצו בסביבת העבודה של google colab.

העבודה על הפרויקט כללה עבודה והעמקה של מספר תחומים כמו –עבודה עם מסד נתונים גדול, תחום עיבוד אותות שמע, תחום הבינה המלאכותית ולמידה עמוקה, ויישום התיאוריה בסביבת העבודה של python וכלל הספריות והחבילות שבה. בנוסף גם כללה עבודה עם ממשק גרפי למשתמש GUI שישמש ככלי להפעלת המערכת הכוללת. הידע שלי הורחב בכלל התחומים הללו. בנוסף לידע התאורטי הפרויקט נתן לי כלים ללמידה עצמאית ולהתמודדות עצמאית בכלל הבעיות והאתגרים ההנדסיים שחווייתי (מפורטים בספר), למדתי על תהליך בניית מערכת מורכבת, על חשיבות התכן לפני המימוש ועוד.

העבודה על הפרויקט הייתה מעניינת ומאתגרת מאוד.

7.4. עמידה בדרישות :

מס'	דרישת המערכת	עמידה בדרישה
1	ממשק גרפי (GUI) נוח להפעלת המערכת.	בוצע
2	יכולת קליטת קטע האודיו מהמיקרופון למחשב באמצעות הממשק הגרפי למשתמש.	בוצע
3	יכולת לסנן רעשים לקבלת אות אודיו נקי של הקטע הנשמע.	בוצע
4	יכולת לזהות ולהבחין בין אירועי קול (קולות של חיה) לבין שקט ולחלץ את האירוע מהקטע הנשמע (מעבר להצעה).	בוצע
5	יכולת הוצאת features (מאפיינים) של קטעי הקול השונים.	בוצע
6	יכולת השוואה של מאפייני קטע הקול מול מאגר קיים	בוצע
7	יכולת שמירת הקולות והמאפיינים שלהם למאגר	בוצע
8	יכולת זיהוי של קולות החיות ב-75% דיוק או יותר עבור 5 חיות לפחות.	בוצע
9	יכולת לספק מידע תמציתי על החיה.	בוצע
10	יכולת לאמן מחדש את המערכת (בדאטא נוסף)	בוצע

טבלה 2 עמידה בדרישות

7.5. הצעות לשיפור :

מימוש המערכת בסביבות פיתוח נוספות:

ניתן לממש את המערכת לזיהוי חיות לפי הקלטות קוליות שלהן בטלפונים החכמים על ידי המרה לקוד המתאים בסביבת העבודה המתאימה לטלפונים החכמים.

7.6. תרומת הפרויקט:

פרויקט זה יכול לתרום המון לאנשים שאוהבים לטייל בטבע, בשדות, וביערות השונים. כאשר המטיילים ישמעו קול של חיה והם לא יזהו אותה, אל להם להיכנס לפאניקה, אלא באמצעות האפליקציה להקליט את הרעש שהחיה משמיעה, ובזמן אמת לקבל מידע עליה ועל התנהגות מומלצת בקרבתה. בנוסף לכך המערכת תוכל להתפתח על ידי הקלטות נוספות שהמטיילים יזינו למערכת, והמערכת תהפוך לאמינה ורחבה יותר ויותר.

7.7. מסקנות:

1. המערכת עומדת בדרישות הנדרשות ועובדת בצורה תקינה לחלוטין ללא בעיות.
2. הסדר והארגון במהלך ביצוע הפרויקט תרמו לעבודה יעילה ונכונה יותר ולבסוף ללמידה מעמיקה יותר על החומר התיאורטי שהקיף הפרויקט.
3. העבודה על הפרויקט כללה עבודה והעמקה של מספר תחומים כמו – תחום עיבוד אותות שמע, תחום הבינה המלאכותית ולמידה עמוקה, ויישום התיאוריה בסביבת העבודה של python וכלל הספריות והחבילות שבה ובנוסף גם עבודה עם ממשק גרפי למשתמש GUI.

אכן הפרויקט חיזק את הבנתי בתחומים של עיבוד אותות , למידה עמוקה , ופיתוח
בpython .

4. באופן כללי העבודה עם סביבת העבודה של python מאוד מעניינת והשימוש באמצעים
שקיימים בה על מנת לממש פתרון חדש לבעיה או לשפר פתרון קיים אתגרה אותי וגרמה לי
להעמיק את הידע שלי ולשפר את המקצועיות שלי.

8. סימוכין

- Wikipedia
- Keras documentation
- Librosa documentation
- Tensorflow tutorial

9. נספחים

ב-disk-on-key המצורף לספר הפרויקט נמצאים : הצעת הפרויקט , ספר הפרויקט והקודים
הרלוונטים לפרויקט זה (לא בהכרח בגרסתם הסופית).