

**המחלקה להנדסת תוכנה**

## **שם הפרויקט: אפליקציה לזיהוי ציפורים**

### **Statement Of Work – הגדרת הפרויקט**

**שמות הסטודנטים:** סתיו סוזן רבינוביץ' 208661090

ליאור מטרני 205478258

**שם המנחה:** ד"ר שרון ילוב-הנדזל

**אישור המנחה על** הגשת הצעת הפרויקט של סתיו וליאור מאושרת

**בדיקת המסמך**

בהצלחה, שרון

**ואישורו להגשה:**

**תאריך ההגשה:** 1/10/2021

### 3. תוכן עניינים

|    |   |
|----|---|
| 2  | 3. תוכן עניינים   |
| 6  | 4. מבוא   |
| 7  | 5. מטרות הפרויקט  |
| 7  | 5.1. תיאור המטרות   |
| 7  | 5.2. יעדים פונקציונליים   |
| 7  | 5.3. מדדים  |
| 8  | 6. דרישות ראשוניות מהמערכת (High Level)   |
| 8  | 6.1. יעדי המערכת  |
| 8  | 6.2. דרישות פונקציונליות  |
| 8  | 6.2.1. דרישות מהאלגוריתם  |
| 8  | 6.2.2. דרישות מהמערכת   |
| 8  | 6.2.3. דרישות מהיישומון   |
| 9  | 6.3. דרישות לא פונקציונליות   |
| 9  | 6.3.1. דרישות עיצוב   |
| 9  | 6.3.2. דרישות ביצועים   |
| 9  | 6.3.3. דרישות הפעלה   |
| 9  | 6.3.4. דרישות תחזוקה  |
| 9  | 6.3.5. דרישות אמינות  |
| 10 | 7. סקירת ספרות ראשונית  |
| 11 | 7.1. מאמר 1: Bird Species Identification from an Image [1]  |
|    | 7.2. מאמר 2: Look Closer to See Better: Recurrent Attention Convolutional Neural Network for Fine-grained Image Recognition [2] |
| 13 | 7.3. מאמר 3: Analyzing the Performance of Apps Developed by using Cross-Platform and Native Technologies [3]                    |
| 14 | 7.1. מאמר 4: CNN BASED CLASSIFIER FOR IDENTIFICATION OF CANINE BREEDS [4]   |
| 15 | 8. תרשים בלוקים   |
| 17 | 8.1. הצגת תרשים בלוקים  |
| 17 | 8.1.1. מאגר מידע ציפורים (Birds Data DB)  |
| 17 | 8.1.2. מאגר משתמשים (Users DB)  |
| 17 | 8.1.3. מאגר ציפורים מזוהות (Recognized Birds DB)  |
| 17 | 8.1.4. מודול הרשמה (Sign-up)  |
| 17 | 8.1.5. מודול התחברות (Sign-in)  |
| 17 | 8.1.6. מודול ראשי (Main Page)   |

|    |   |
|----|---|
| 18 | 8.1.8. מודול זיהוי משתמש (User's Recognitions)                            |
| 18 | 8.1.9. מודול חיפוש ציפור (Search Bird)                                    |
| 18 | 8.1.10. מודול תוצאת חיפוש (Presenting Results)                            |
| 18 | 8.1.11. מודול מיקומי ציפור (Bird's Locations)                             |
| 18 | 8.1.12. מודול זיהוי לפי תמונה (Recognition By Image)                      |
| 18 | 8.1.13. אלגוריתם לזיהוי ציפורים (Birds Recognition Algorithm)             |
| 18 | 8.1.14. מודל תוצאת אלגוריתם (Recognition Result)                          |
| 18 | 8.1.15. מודול אימות זיהויים (Recognition Validation)                      |
| 18 | 8.2. התממשקות עם רכיבים חיצוניים  |
| 19 | 8.3. הבהרת גבולות הפיתוח  |
| 19 | 8.4. תיאור לב הפתרון  |
| 19 | 8.5. תיאור הרכיבים העיקריים במערכת  |
| 20 | 9. ניתוח פונקציונלי ראשוני  |
| 21 | 9.1. חיפוש ציפור  |
| 21 | 9.1.1. זרימה ראשית  |
| 22 | 9.1.2. זרימה משנית א  |
| 23 | 9.2. זיהוי ציפורים  |
| 23 | 9.2.1. זרימה ראשית  |
| 23 | 9.2.2. זרימה משנית א  |
| 24 | 9.3. אימות צפר מקצועי   |
| 24 | 9.3.1. זרימה ראשית  |
| 25 | 10. הצגה ראשונית של חלופות למימוש   |
| 25 | 10.1. אלגוריתם לזיהוי ציפור   |
| 25 | 10.1.1. פיתוח באמצעות שימוש באלגוריתמים מבוססי תכונות התמונה              |
| 25 | 10.1.2. פיתוח באמצעות שימוש באלגוריתמים מבוססי ראייה ממוחשבת ולמידה עמוקה |
| 25 | 10.1.3. פיתוח באמצעות שילוב האלגוריתמים                                   |
| 26 | 10.1.4. השוואת החלופות  |
| 26 | 10.1.5. החלופה הנבחרת   |
| 27 | 10.2. דרכי פיתוח האלגוריתם  |
| 27 | 10.2.1. פיתוח אלגוריתם באמצעות Python                                     |
| 27 | 10.2.2. פיתוח אלגוריתם באמצעות Matlab                                     |
| 27 | 10.2.3. פיתוח בסביבת R  |
| 27 | 10.2.4. פיתוח אלגוריתם באמצעות C++  |
| 28 | 10.2.5. השוואת החלופות  |
| 28 | 10.2.6. החלופה הנבחרת   |

|        |   |    |
|--------|---|----|
| 10.3   | מימוש פלטפורמת SDK  | 29 |
| 10.3.1 | פיתוח באמצעות Native Android                                  | 29 |
| 10.3.2 | פיתוח באמצעות Native IOS                                      | 29 |
| 10.3.3 | פיתוח בפלטפורמת React Native                                  | 29 |
| 10.3.4 | פיתוח בפלטפורמת Flutter                                       | 29 |
| 10.3.5 | השוואת החלופות  | 30 |
| 10.3.6 | החלופה הנבחרת   | 30 |
| 11     | סקר שוק ראשוני  | 31 |
| 11.1   | יישומון Merlin Bird ID [5]                                    | 31 |
| 11.1.1 | יתרונות:  | 31 |
| 11.1.2 | חסרונות:  | 31 |
| 11.2   | אתר BirdNET [6]   | 31 |
| 11.2.1 | יתרונות:  | 31 |
| 11.2.2 | חסרונות:  | 31 |
| 11.3   | אתר הצפרות הישראלי [7]  | 32 |
| 11.3.1 | יתרונות:  | 32 |
| 11.3.2 | חסרונות:  | 32 |
| 11.4   | לקסיקון מפה: הציפורים - המדריך השלם לציפורי אירופה וישראל [8] | 32 |
| 11.4.1 | יתרונות:  | 32 |
| 11.4.2 | חסרונות:  | 32 |
| 11.5   | השוואה  | 33 |
| 12     | האמצעים/ הכלים הנדרשים  | 34 |
| 12.1   | התייחסות לסביבת הפיתוח וסביבת ההרצה                           | 34 |
| 13     | תוצרי הפרויקט   | 35 |
| 13.1   | תיאור תוצרי הפרויקט   | 35 |
| 13.1.1 | אלגוריתם לזיהוי ציפור לפי תמונה                               | 35 |
| 13.1.2 | אלגוריתם חיפוש ציפורים לפי אזור                               | 35 |
| 13.1.3 | פיתוח יישומון למכשיר נייד לזיהוי וחיפוש הציפורים              | 35 |
| 13.2   | תיאור התוצרים בהתייחס למערכת הסופית                           | 35 |
| 13.3   | שלבי הפיתוח והבדיקות  | 35 |
| 14     | תוכנית עבודה ראשונית לביצוע הפרויקט                           | 36 |
| 15     | פערים   | 37 |
| 15.1   | פערי ידע  | 37 |
| 15.2   | זמינות הציוד הנדרש  | 37 |
| 16     | סיכונים עיקריים ודרכי התמודדות                                | 38 |

|       |  |    |
|-------|--|----|
| 16.1. | זיהו לא נכון של ציפור קיימת באלגוריתם (20)                 | 38 |
| 16.2. | בעיית התממשקות עם מסד נתוני המערכת (15)                    | 38 |
| 16.3. | אי התאמת המערכת לכלל המכשירים החכמים (9)                   | 39 |
| 16.4. | אי סנכרון בין מפות ותוצאות אלגוריתמים של משתמשים שונים (9) | 39 |
| 16.5. | בעיות התממשקות עם מערכת מפות חיצונית (8)                   | 39 |
| 16.6. | חלוקת עבודה לא יעילה (6)                                   | 40 |
| 16.7. | ריצה איטית של האלגוריתם (6)                                | 40 |
| 16.8. | אי מציאת ציפור בחיפוש (3)                                  | 40 |
| 17.   | רשימת מקורות (ראשונית)                                     | 41 |

#### 4. מבוא

האזור הגיאוגרפי של ישראל ממוקם במעבר בין שלוש יבשות: אסיה, אירופה ואפריקה. לכן, במהלך השנה (ובעיקר בעונות הנדידה) ניתן לצפות במגוון ציפורים בעת נדידתן, עוברות מעל שמי הארץ.

כיום, בשביל הצפר החובב, עיקר המערכות המעקב אחרי ציפורים בצורה מקצועית הינן באתרי צפרות ברחבי הארץ וזיהוי של הציפורים נעשה על ידי צפרים מנוסים. לרוב הצפרים החובבים או אנשים המתעניינים, אין גישה מתמדת לאתרי צפרות ואין אפשרות להתייעץ או לשאול צפר מנוסה בכל רגע נתון. כך נוצר מצב שהצפר החובב לא תמיד ידע לזהות את הציפור הנמצאת לידו – ובאותה מהירות שהציפור נמצאה, היא יכולה לעוף.

כמוכן, באתרי הצפרות, עיקר ההתייחסות היא לציפורים באותו האזור, אך הצפרים המקצועיים אינם נמצאים בכל מקום ובכל זמן. מה שמונע מהם לעקוב בכל זמן נתון אחר מיקומי הציפורים.

מטרת הפרויקט היא לתת מענה לחובבי צפרות וגם לצפרים ברחבי הארץ על ידי שימוש ביישומון שיודע לזהות ציפורים מתוך תמונה, וגם מאפשר לשמור את המיקום של אותה ציפור במאגר.

מעבר לכך, משתמשי היישומון יוכלו לראות אילו ציפורים נמצאות באזור שלהם ולחפש מידע על ציפורים שונות. בדרך זו, זיהוי הציפורים יהיה זמין לכלל המתעניינים וכן צפרים ואתרי צפרות יוכלו לקבל מידע נרחב על מיקומי ציפורים ולבצע מעקב נרחב יותר.

ביישומון יעשה שימוש באלגוריתם מבוסס ראייה ממוחשבת ורשתות נוירונים שידע לזהות את הציפור מהתמונה, וגם ישמור את נתוני הציפור לצורך אימונים נוספים שישפרו את איכות הזיהוי.

הפרויקט יתוכנן לשלושה חלקים - פיתוח אלגוריתם לזיהוי הציפורים, צד לקוח וצד שרת.

- פיתוח האלגוריתם - עיקר המערכת, באמצעות רשתות נוירונים קיימות וספריות ראייה ממוחשבת, נפתח אלגוריתם שבסופו יקבל תמונה של ציפור וידע להחזיר למשתמש מידע אודות אותה הציפור.
- צד לקוח וחווית המשתמש - בחלק זה נפתח את היישומון עצמו של המערכת, בחלק זה נתרכז בחלונות אותם נגיש למשתמש, נשים דגש על עיצוב ממשק וחווית שימוש של המשתמש.
- צד שרת ובסיס הנתונים - בחלק זה הקשר בין צד הלקוח לאלגוריתם, נחבר בין היישומון למאגרי הנתונים ונפתח פונקציות נוספות אותן נרצה להציג ללקוח, בהן נשלב שימוש במאגרי הנתונים הקיימים לנו.

ביצוע העבודות בכל שלושת החלקים יבוצעו על ידי שני הסטודנטים בסנכרון מלא, אך לחלק מהתחומים יהיה סטודנט האחראי למעקב וניהול המשימות באותו התחום. חלוקת תחומי האחריות בפרויקט תותאם לפי החוזקות של כל אחד, הידע הקיים הרלוונטי ותחומי העניין של כל אחד מחברי הפרויקט.

סתיו סוזן רבינוביץ - התמחות בינה מלאכותית, אחראית על צד לקוח.

ליאור מטרני - התמחות בינה מלאכותית, אחראי על צד שרת.

פיתוח האלגוריתם עבור המערכת יהיה באחריות שניהם ובשיתוף פעולה מלא.

## 5. מטרות הפרויקט

### 5.1. תיאור המטרות

מטרת הפרויקט היא לתת מענה לחובבי צפרות וגם לצפרים ברחבי הארץ על ידי שימוש ביישומון, המאפשר לזהות ציפור על ידי תמונה, לקבל מידע עליה ועל המיקום של אותה ציפור במאגר. היישומון יאפשר לשתף נתונים על ציפור בין צפרים חובבים, מבקרי אתרי צפרות או מתעניינים.

### 5.2. יעדים פונקציונליים

1. כתיבת אלגוריתם ראייה ממוחשבת לזיהוי סוג הציפור לפי תמונה.
2. הקמת יישומון סלולרי ייעודי לזיהוי התמונות ע"י מימוש האלגוריתם.
3. הצגה ביישומון של ציפורים באזור לפי מיקום.
4. הצגת מידע ביישומון מידע אודות הציפור שזוהתה או אודות ציפור לפי בקשת משתמש.

### 5.3. מדדים

על מנת לבחון את האלגוריתם, יוקם מאגר תמונות שצולמו ממכשיר חכם (וניתן בוודאות לזהות את הציפור) ומתמונות של ציפורים מהרשת, ובאמצעות תמונות אלה ייבחנו אחוזי ההצלחה של אלגוריתם הקלאסיפיקציה:

1. לפחות 60% מהציפורים יזוהו לפי סוג הציפור.
2. לפחות 75% מהציפורים יזוהו לפי משפחת הציפורים (יוניים, עגוריים, קוקייתיים וכו'..).
3. זיהוי ואימון האלגוריתם על לפחות 6 ציפורים מתוך 3 משפחות ציפורים שונות.
4. ציון של 3 ומעלה (מתוך 5) בסקר שביעות רצון של המשתמשים ביישומון.

## 6. דרישות ראשוניות מהמערכת (High Level)

### 6.1. יעדי המערכת

כפי שתואר בפרק 5, מטרת הפרויקט היא פיתוח אלגוריתם היודע לזהות ציפורים מתוך תמונה ובניית יישומן המאפשר להשתמש באלגוריתם שפיתחנו, וגם מאפשר שמירת מיקום הציפור שהאלגוריתם זיהה במאגר.

### 6.2. דרישות פונקציונליות

#### 6.2.1. דרישות מהאלגוריתם

6.2.1.1. האלגוריתם ידע לזהות את סוג הציפור ואת משפחת הציפורים

6.2.1.2. האלגוריתם ילמד ויתאמן על תמונות חדשות שנוספות למאגר על ידי המשתמשים.

#### 6.2.2. דרישות מהמערכת

6.2.2.1. תקשורת בין השרת המכיל את האלגוריתם ומאגר הנתונים למכשיר עליו מופעל היישומן.

6.2.2.2. אימון האלגוריתם, עיבוד התמונה והזיהוי יתבצעו על השרת ולא במכשיר המשתמש.

6.2.2.3. המערכת תדע להעביר למאגר הנתונים את פרטי התמונה והזיהוי החדשים.

6.2.2.4. המערכת תדע לייבא ממאגר הנתונים זיהויים ונתונים לפי משתמש, לפי שם הציפור ולפי המיקום והרדיוס המבוקשים.

#### 6.2.3. דרישות מהיישומן

6.2.3.1. היישומן יכיל מסך אשר יציג את נתוני המשתמשים והזיהויים שלהם.

6.2.3.2. היישומן מאפשר להציג את התוצאה של הציפור שזוהתה בתמונה.

6.2.3.3. היישומן מאפשר להציג על מפה ציפורים שזוהו בסביבת המשתמש ותדע לסנן גם לפי התאריך.

6.2.3.4. היישומן יאפשר להתחבר בתור משתמש רשום או משתמש שאינו רשום (אורח).

6.2.3.5. היישומן מציג עבור ציפור שזוהתה פרטים נוספים (שם, משפחת ציפורים אליה משתייכת, תמונות נוספות ואזורים בהם נצפתה בחודש האחרון).



## **6.3. דרישות לא פונקציונליות**

### **6.3.1. דרישות עיצוב**

6.3.1.1. ממשק ברור ונוח המאפשר למשתמשים חדשים, ללא רקע טכנולוגי, יכולת לתפעל את היישומון בעצמם מהשימוש הראשון.

6.3.1.2. עיצוב מינימליסטי בכדי למנוע בלבול אצל המשתמש.

### **6.3.2. דרישות ביצועים**

6.3.2.1. היישומון יחזיר תשובה למשתמש על הציפור שזוהתה בזמן מינימלי (יגדר בהמשך).

6.3.2.2. המערכת תוכל לעבוד מול מספר משתמשים במקביל.

### **6.3.3. דרישות הפעלה**

6.3.3.1. היישומון יהיה נגיש למכשיר חכם מסוג Android.

6.3.3.2. מערכת ההפעלה של המכשיר החכם עליו היישומון יופעל תהיה מגרסה 7.0 ומעלה.

6.3.3.3. היישומון יוכל להתחבר למצלמה, לגלריה ולמיקום של המכשיר עליו מותקן.

### **6.3.4. דרישות תחזוקה**

6.3.4.1. למערכת תהיה תאימות עם גרסאות קודמות של האפליקציה.

6.3.4.2. המערכת צריכה להכיל מסד נתונים גדול מספיק בכדי להריץ את האלגוריתם ולהכיל את הנתונים מבלי שהאפליקציה תקרוס.

6.3.4.3. המערכת תוכל לעבוד מול מספר משתמשים במקביל.

### **6.3.5. דרישות אמינות**

6.3.5.1. המערכת תדע לזהות שיש שגיאה בחיבור לשרת ולכן לא מגיעה תשובה מהשרת.

6.3.5.2. למערכת תהיה תאימות עם גרסאות קודמות של האפליקציה.

## 7. סקירת ספרות ראשונית

זיהוי זן הציפור הוא משימה קשה שעדיין עד היום, לעיתים ישנם ויכוחים על סיווג הזיהוי של הציפור.. זאת בעיה קשה שדוחפת את גבולות היכולות החזותיות עבור בני אדם ומחשבים כאחד. למרות שלהרבה ציפורים שונות יש את אותם החלקים (כמו מקור, כנפיים וכדומה), הזנים השונים של הציפורים מאובחנים בעזרת צורת חלקים אלו ובצבעם.

בנוסף, בדקנו את רמות הביצועים של אותם יישומנים שפותחו בטכנולוגיות השונות.

## 7.1. מאמר 1: Bird Species Identification from an Image [1]

המאמר פורסם בשנת 2011 ועוסק בזיהוי ציפורים בעזרת אלגוריתמי למידת מכונה שונים. במאגר ישנם 200 זני ציפורים שונים, 11,778 תמונות שונות שלהם. כך שבממוצע לכל ציפור ישנם 58 תמונות שונות לזיהוי.

המאמר מציג את האלגוריתמים שנבחנו ואת שיטת העבודה של כל אלגוריתם לניתוח התמונה, חיפוש המאפיינים המתאימים (למשל – צורת המקור, צבעי המקור, צורת הכנפיים, גודל וכו'). האלגוריתם שנבחנו הינם –

- סיווג בייסיאני נאיבי (Naïve Bayes) – שיטת סיווג בה ההנחה ה"נאיבית" אומרת כי אין תלות בין תכונות האובייקטים המסווגים כאשר סיווגם ידוע.
- מכונת תמך וקטורי (Support Vector Machines) – טכניקת למידה מונחת לניתוח נתונים, סיווג ורגרסיה. המסווג נוצר על ידי מפריד ליניארי.
- אלגוריתם שכן קרוב (K-nearest Neighbors) -אלגוריתם חסר פרמטרים, כאשר הלימוד בו מבוסס על המופעים בהם הפונקציה מקורבת מקומית.
- ניתוח מפלה ליניארי (Linear Discriminant Analysis(LDA) – בשיטה זו נמצא שילוב ליניארי של תכונות המאפיין או המפריד בין מספר מחלקות או מקרים. לרוב בשימוש כמסווג ליניארי או לצורך הפחתת מימדים.
- עצי החלטה (Decision Trees) – מודל חיזוי, הממפה תצפיות על פריט ויוצר מסקנות על ערך היעד של הפריט. במבנה של עצים אלה, עלים מייצגים סיווגים אפשריים וענפים מייצגים צירופים של תכונות אשר יובילו למחלקות הסיווג.
- אלגוריתם יער אקראי (Random Forests) – שיטת למידה לסיווג ורגרסיה המבוססת על ריבוי עצי החלטה, בהן התוצאה מתקבלת משכלול עצי ההחלטה.
- מסווג אחד מול השאר (One Vs Rest classifier with Logistic Regression) – בשיטה זו ישנו מסווג יחיד לכל מחלקה. כל מסווג בסיס מייצר ערך מספרי להחלטתו בכדי למנוע עמימות.

לאחר ניתוח המאפיינים, העבירו את הנתונים והניתוחים שנעשו לתוך אלגוריתם למידת מכונה, אשר סיווג את הציפורים לפי המאפיינים שהתקבלו.

| Method              | Training Accuracy | Testing Accuracy | Using certainty metric | Using PCA | Using Feature Selection | Using PCA + Feature Selection |
|---------------------|-------------------|------------------|------------------------|-----------|-------------------------|-------------------------------|
| Naive Bayes         | 33.07             | 19.22            |                        |           |                         |                               |
| KNN                 | 45.43             | 31.18            |                        |           |                         |                               |
| Decision Trees      | 99.83             | 24.35            |                        |           |                         |                               |
| Random Forests      | 99.39             | 33.58            |                        |           |                         |                               |
| LDA                 | 63.56             | 45.44            | 46.73                  | 47.81     | 47.7                    | 47.38                         |
| SVM                 | 50.67             | 43.91            | 48.15                  | 48.74     | 49.11                   | 46.93                         |
| Logistic Regression | 84.42             | 51.61            | 52.42                  | 53.31     | 51.02                   | 53.65                         |

טבלה 7.1.1 – טבלה מתוך מאמר [1], תוצאות האלגוריתמים הנבדקים במאמר

התוצאות הנתונות במאמר מצביעות על כך שלאחר חלוקה למאפיינים המשותפים לכלל הציפורים ולאחר שימוש באלגוריתמים הנבחרים, השימוש בשיטת סיווג של עץ החלטה היא השיטה בעלת אחוזי הדיוק הגבוהים ביותר במהלך האימון, עם אחוז דיוק של 99.83%, אך כאן ניתן לראות שבשיטה זאת מתקיים Over fitting משום שבמהלך המבחן ישנם רק 24.35%, אך בבחינה של השיטות – השיטה בעלת תוצאת הבחינה הגבוהה ביותר ללא Over Fitting - הינה רגרסיה לוגיסטית, עם אחוז דיוק של 53.65 במהלך יישומו יחד עם בחירת תכונות מסוימות (הורדו מספר תכונות שגם כן זיהוי שאינן עזרו לתוצאה, או אפילו פגועה בה).

ניתן לראות את אלגוריתם ה-LDA שאצלו במהלך המבחן מול קבוצת הולידציה הגיע לציון של 63.56% אך ברגע שהריצו אותו על קבוצה המבחן, הוא ירד בדיוק ברמה של 18% מה שיכול להעיד גם כן על התאמת יתר לקבוצת האימון.

אלגוריתמים אלו אינם הגיעו לתוצאה גבוהה כמו שציפו החוקרים, ואף במספר מקרים הגיעו למצב של התאמת יתר.

## 7.2. מאמר 2: Look Closer to See Better: Recurrent Attention Convolutional

### [2] Neural Network for Fine-grained Image Recognition

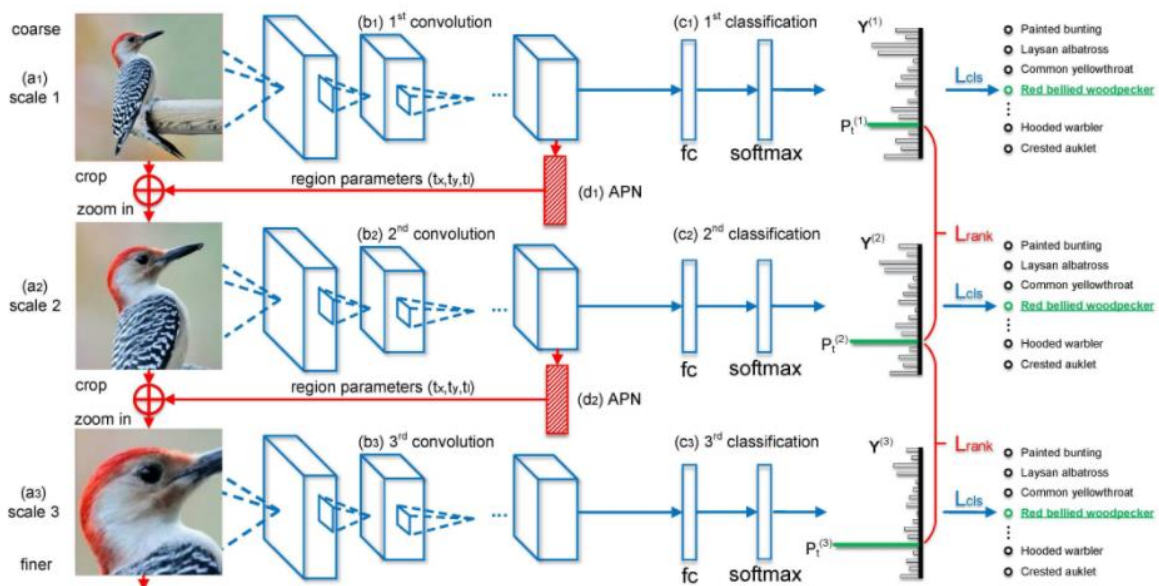
המאמר [2] פורסם בשנת 2017 ומציג את הדרך בה ניתן בעזרת רשת נוירונים לקבל תמונה אחת, ולהוציא ממנה מספר רב של נתונים על ידי רשת נוירונים מסוג RA-CNN.

רשת מסוג RA-CNN מקבלת קלט אחד, לאחר מכן מתבצע פיצול של התמונה לפי המאפיינים אותם הוא מזהה, הרשת מנתחת את אותו המאפיין ולאחר מכן הרשת מציגה את הניתוח הסופי של התמונה אותה קיבלה.

שיטת העבודה באלגוריתם זה הנה:

- קבלת תמונה.
- התמקדות על המאפיינים אותם הרשת תרצה לנתח.
- ניתוח כל מאפיין בנפרד.
- חיזוי התוצאה.

ניתן לראות באיור 7.2.1 את הדרך בה רשת הנוירונים מקבלת תמונה מלאה של ציפור ומתמקדת בראש הציפור על מנת לנתח את המאפיינים הרלוונטיים עבורה.



איור 7.2.1 – תמונה מתוך מאמר [2], ניתוח תמונה ברשת הנוירונים

### 7.3. מאמר 3: Analyzing the Performance of Apps Developed by using Cross-Platform and Native Technologies [3]

במאמר זה, בוחנים את רמת הביצועים של פיתוח אפליקציות בכל סביבת עבודה.

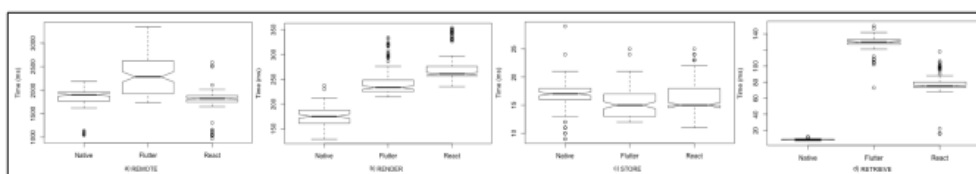
המתחרים במאמר זה אלו הסביבות:

- "ריאקט נייטיב" (React Native) – סביבת פיתוח לאנדרואיד ו־IOS יחדיו, לא צריך לפתוח פרויקט מיעוד לכל סביבה וכותבים הכל בשפה משותפת. בסביבה זו כותבים בשפת JavaScript וספרייה זו פותחה על ידי חבר Facebook.
- "פלאטר" (Flutter) – סביבת פיתוח לאנדרואיד ו־IOS יחדיו, לא צריך לפתוח פרויקט מיעוד לכל סביבה וכותבים הכל בשפה משותפת. בסביבה זו כותבים בשפת Dart, שפה זו פותחה על ידי Google.
- פיתוח למערכת הפעלה מסוג IOS על ידי שימוש ב"סוויפט" (Swift) – סביבת פיתוח רק למכשירי IOS. הסביבה פותחה על ידי חברת Apple. בסביבה זו אי אפשר לפתח גם למכשירי Android.
- פיתוח למערכת הפעלה מסוג Android על ידי שימוש בשפת "ג'אווה" (Java) – סביבת פיתוח רק למכשירי Android, בשפה זו אי אפשר לכתוב פרויקט גם למכשירי IOS.

במאמר זה מחלקים את המבחנים לשני ייעודים שונים:

#### 1. פיתוח לסביבת Android – בה קיימת השוואה בין React Native, Flutter, Java.

- החוקרים בחנו את זמני התגובה של היישומים אשר פותחו בדרכים שונות לצורך ביצוע אותן פעולות למכשירים בעלי מערכת הפעלה Android. תוצאות המסקנות הן סטטיסטיות לפי הפעולות שבוצעו.
- החוקרים הגיעו למסקנה בה שכשמדובר על פונקציות הקשורות לפעולות במערכת ההפעלה, Java הייתה בעלת התוצאות הטובות ביותר, אך פיתוח ב־Java מותאם אך ורק למערכת הפעלה זו ואינה מותאמת למערכת הפעלה מסוג IOS.
- מסקנה נוספת אליה הגיעו החוקרים הייתה שכאשר בוחנים בקשות מול שרת או אחסון בתוך המכשיר הנייד, כלל היישומים מכלל סביבות הפיתוח הגיעו לאותו זמן תגובה.
- את תוצאות ההשוואות מהמאמר ניתן לראות באיור 7.3.1



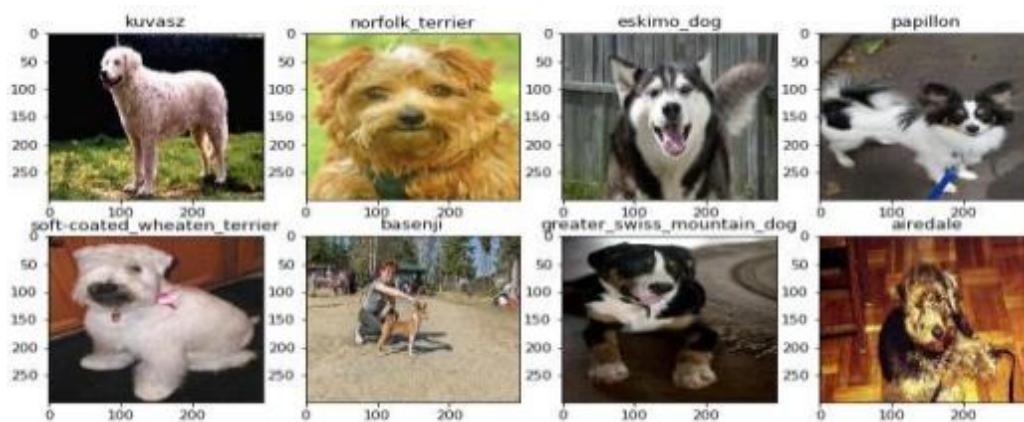
איור 7.3.1 – תמונה מתוך מאמר [3], השוואה סטטיסטית בין כלל השיטות עבור Android

## 7.1. מאמר 4: CNN BASED CLASSIFIER FOR IDENTIFICATION OF CANINE

### [4] BREEDS

במאמר זה מתבצע זיהוי גזע הכלב לפי תמונה. בעיית זיהוי גזע הכלב הנה בעיית זיהוי לא קלה, זאת משום קיימת כמות רבה של גזעי כלבים שונים, וגם – יש גזעים שונים של כלבים מאד דומים במבנה, צורה, גודל וצבע וקיימים הבדלים מעטים הלא נראים לעין בין גזעי כלבים, ולעיתים הרבה הבדלים ושנראים לעין.

- החוקרים השתמשו במאגר של 8351 תמונות של 133 סוגי גזעים שונים של כלבים. לאימון המודל החליטו החוקרים לחלק את המידע כך שלאימון ישנם 80% מכלל התמונות (6680 תמונות) לקבוצת האימות ישנם 10% מכלל התמונות (836 תמונות) ולקבוצת המבחן ישנם 10% מכלל התמונות. תמונות אלו של הכלבים נלקחו ממאגר הנתונים ImageNet, Flickr, Google Image. תמונות אלו הכילו 120 גזעים שונים של כלבים.
- ImageNet הוא מאגר מידע חזותי גדול המיועד לשימוש במחקר תוכנות לאובייקטים חזותיים, כמו לדוגמא, המאגר גם כן משמש לזיהוי סוגי כלבים. במאגר זה ישנו מעל 14 מיליון תמונות שהעבור ביאור כדי לציין אילו אובייקטים מופיעים בתמונות אלו. ישנם במאגר זה קטגוריות לדוגמא כמו "בלון" או "תות שדה" המוכבות מכמה מאות תמונות.
- החוקרים לקחו את מאגר התמונות והמירו את כל התמונות שהיו בגדלים וברזולוציות שונות לגודל של 224X224 כדי שיעבוד עם רשתות כמו VGG16, ResNe50, MobileNet, InceptionV3 וגם לגודל של 229X229 בשביל Xception.



איור 7.4.1 – תמונה מתוך מאמר [4], תמונות לדוגמה של גזעי כלבים

- מאגר זה היה יחסית קטן משום שישנם 133 גזעים שונים של גזעים וכך יצא שלגזע בעל מספר התמונות הקטן ביותר היו 33 תמונות ולגזע בעל מספר התמונות הגדול ביותר היו 96 תמונות.
- לאחר בניית מאגר הנתונים, לקחו רשת CNN שאינה מאומנת כלל ואימנו אותה על מאגר התמונות. כתוצאה מאימון התמונות על מאגר זה החוקרים הגיעו לדיוק של 49.46% בקבוצת האימות ול47.42% של דיוק בקבוצת המבחן.
- לאחר התוצאה היחסית הנמוכה שיצאה במקצה זה, החוקרים חילקו את הניסוי הבא לשני חלקים. בחלק הראשון, בנו החוקרים רשתות נוספות: VGG16, InceptionV3, Xception לא מאומנות והריצו אותם על המאגר השלם של ImageNet ששם משתמשים בהוצאת "פיצ'רים" של כל תמונה, שיטה שונה ממה שנעשה בעזרת רשת הCNN. לאחר מכן

החוקרים לקחו את הרשתות שידעות עכשיו להוציא "פיצ'רים" והריצו אותה על קבוצת האימון, הולידציה והמבחן של המאגר נתונים של החוקרים. את נתונים אלו החוקרים שמרו. החוקרים רק רצו את החלק של זיהוי החלקים הספציפים שניתן מרשתות אלו. כך עשו החוקרים לשלושת הרשתות. בחלק השני, החוקרים לקחו את המודל המאומן שהיה לכל רשת כזו ועל הפלט שלו החוקרים הוסיפו עוד רשת על מנת לזהות רק את גזע הכלב ולא את ה"פיצ'רים" שמקבלים מרשתות אלו. לאחר מכן החוקרים שוב אימנו את המודלים המשולבים שבנויים מזיהוי ה"פיצ'רים" ולאחר מכן בזיהוי סוג הכלב ובדקו את התוצאות.

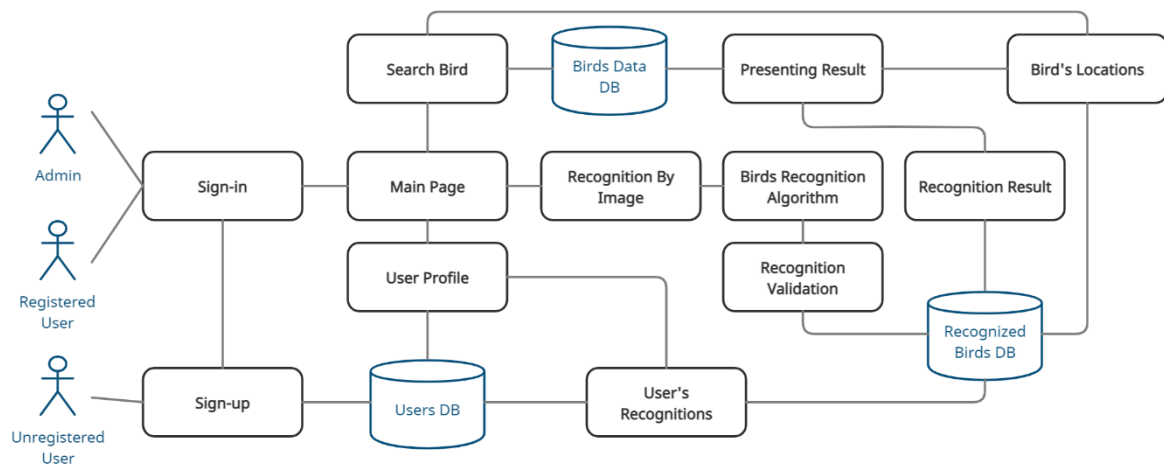
| Pre-trained architecture ▾ | Validation ▾ | Test ▾ |
|----------------------------|--------------|--------|
| VGG16                      | 54.76%       | 59.89% |
| Xception                   | 83.71%       | 84.97% |
| InceptionV3                | 76.86%       | 79.78% |

איור 7.4.2 – תמונה מתוך מאמר [4], תמונות לדוגמה של גזעי כלבים

ניתן לראות שתוצאות אלו הגיעו לציונים הרבה יותר גבוהים מהרשתות מהניסוי הקודם וניתן לראות שהרשת בעלת הציון הגבוה ביותר הייתה הרשת Xception שהגיעה לציון של 83.71% בקבוצת האימון ו84.97% בקבוצת המבחן. החוקרים הגיעו למסכנה של רשת הבנויה בעזרת Xception מבוססת על רשת CNN עם מודלים שנעשו בעזרת Transfer learning בעלת הציון הגבוה ביותר כאשר מנסים לזהות גזעים של כלבים.



## 8. תרשים בלוקים



איור 8.1 – תרשים בלוקים המתאר את ארכיטקטורת המערכת

### 8.1. הצגת תרשים בלוקים

באיור 8.1 ניתן לראות את תרשים הבלוקים המתאר את ארכיטקטורת המערכת

#### 8.1.1. מאגר מידע ציפורים (Birds Data DB)

מאגר אשר יכיל את כל המידע על הציפורים שיופיע למשתמש כאשר יחפש אותן.

#### 8.1.2. מאגר משתמשים (Users DB)

המאגר בו ישמרו הנתונים אודות כל המשתמשים הרשומים במערכת, כלומר המשתמשים הרגילים, מנהלים.

#### 8.1.3. מאגר ציפורים מזוהות (Recognized Birds DB)

מאגר מיקומי הציפורים שזוהו במערכת. ממאגר זה ילקחו התמונות לאימון האלגוריתם כל תקופה.

#### 8.1.4. מודול הרשמה (Sign-up)

מסך הרשמה למערכת, כל משתמש רגיל ולא רשום יכול להירשם למערכת משם.

#### 8.1.5. מודול התחברות (Sign-in)

מסך התחברות למערכת, משם יתחברו למערכת גם משתמשים רגילים וגם מנהלים.

#### 8.1.6. מודול ראשי (Main Page)

המסך הראשי, ממנו יהיו התפריטים לכלל האפשרויות ביישומן.

#### 8.1.7. מודול פרופיל משתמש (User Profile)

מסך שבו המשתמש יכול לראות את כל ההגדרות שלו, לשנות, לעדכן, ולמחוק את המשתמש.

### 8.1.8. מודול זיהויי משתמש (User's Recognitions)

מסך בו יוצגו כל הזיהויים שהמשתמש עושה ביישומן שלו.

### 8.1.9. מודול חיפוש ציפור (Search Bird)

מודול זה יבקש מהמשתמש איזה ציפור הוא רוצה לחפש או מה רדיוס החיפוש. במידה והמשתמש יחליט לחפש שם של ציפור, המערכת תעביר ממאגר הנתונים את המידע עבור הציפור אותה המשתמש מבקש למודול תוצאת החיפוש. במידה והמשתמש החליט להעביר רדיוס, המערכת תעביר למודל מיקומי ציפור.

### 8.1.10. מודול תוצאת חיפוש (Presenting Results)

במסך זה יוצג למשתמש מידע אודות הציפור שעלתה בחיפוש או בתוצאה.

### 8.1.11. מודול מיקומי ציפור (Bird's Locations)

מסך שיראה את כל המיקומים שנראו לאחרונה של ציפור נבחרת, המערכת תציג מספר תוצאות אחרונות שהיו להצגת הציפור.

### 8.1.12. מודול זיהוי לפי תמונה (Recognition By Image)

מסך בו תהיה אפשרות לראות את המצלמה, לבחור לצלם את התמונה ואם התמונה לא יצאה ברורה ישנה אפשרות לצלם מחדש, במסך זה תהיה את האפשרות לשלוח את התמונה לתהליך הזיהוי של התמונה לפי האלגוריתם.

### 8.1.13. אלגוריתם לזיהוי ציפורים (Birds Recognition Algorithm)

האלגוריתם אותו נבנה לצורך זיהוי הציפורים. אלגוריתם זה הוא חלק מתוצרי הפרויקט.

### 8.1.14. מודל תוצאת אלגוריתם (Recognition Result)

במודול זה נקבל את תוצאת הזיהוי, פרטי הזיהוי יועברו למאגר המיקומים והמודל יעביר את המשתמש למסך הצגת התוצאה.

### 8.1.15. מודול אימות זיהויים (Recognition Validation)

כדי לאפשר לציפורים שזוהו ותמונותיהן להצטרף למאגר האימון, מנהל המערכת יוכל באמצעות מודול זה לאמת תוצאות ולהעביר את נתוניהם למאגר הציפורים של האלגוריתם.

## 8.2. התממשקות עם רכיבים חיצוניים

### 8.2.1. מערכת מפות

נשתמש בפלטפורמה קיימת של מפות, עליה נציג את המיקומים.

### 8.2.2. אחסון מאגרי הנתונים והאלגוריתמים

המידע על המשתמשים, הזיהויים, הציפורים וגם אימון האלגוריתם והרצתו יאוחסנו ובוצעו על שרת חיצוני. באמצעות מאגר זה נוכל לעקוב אחרי התהליכים המתרחשים במערכת.

### 8.2.3. מצלמת המשתמש וגלריית תמונות המשתמש

המידע על תמונות המשתמש או שימוש במצלמת המשתמש במכשיר החכם הכרחית לצורך קבלת תמונה לזיהוי עבור המערכת. רק ע"י התממשקות למצלמה או לגלריה, המשתמש יכול להעלות תמונה לזיהוי.

### **8.3. הבהרת גבולות הפיתוח**

אנחנו לא נפתח את המפה אלא נשתמש בפלטפורמה להצגת מפה שכבר קיימת.

### **8.4. תיאור לב הפתרון**

המערכת מאפשרת לצפרים וחובבי צפרות דרך נוחה וקלה לזהות ציפורים אותן הם רואים בסביבתם על ידי שימוש באלגוריתם מבוסס ראייה ממוחשבת ורשת ניורונים ובעזרת ממשק נוח ופשוט להבנה, כך שאין צורך בידע טכנולוגי מקדים כדי להשתמש בו.

### **8.5. תיאור הרכיבים העיקריים במערכת**

#### **8.5.1. רכיבים עיקריים**

המערכת תמומש על ידי יישומון סלולרי למכשיר חכם (תומך מערכת הפעלה של Android), אותו המשתמש יכול להתקין במכשיר החכם שלו. היישומון יכלול את ממשקי האינטראקציה עם המשתמש, אשר יעזרו למשתמש לזהות את אותה הציפור.

בנוסף, תהיה למשתמש אפשרות להנות מפונקציות נוספות במערכת כמו מציאת ציפורים שזוהו באזור שלו, חיפוש ציפורים וכו'.

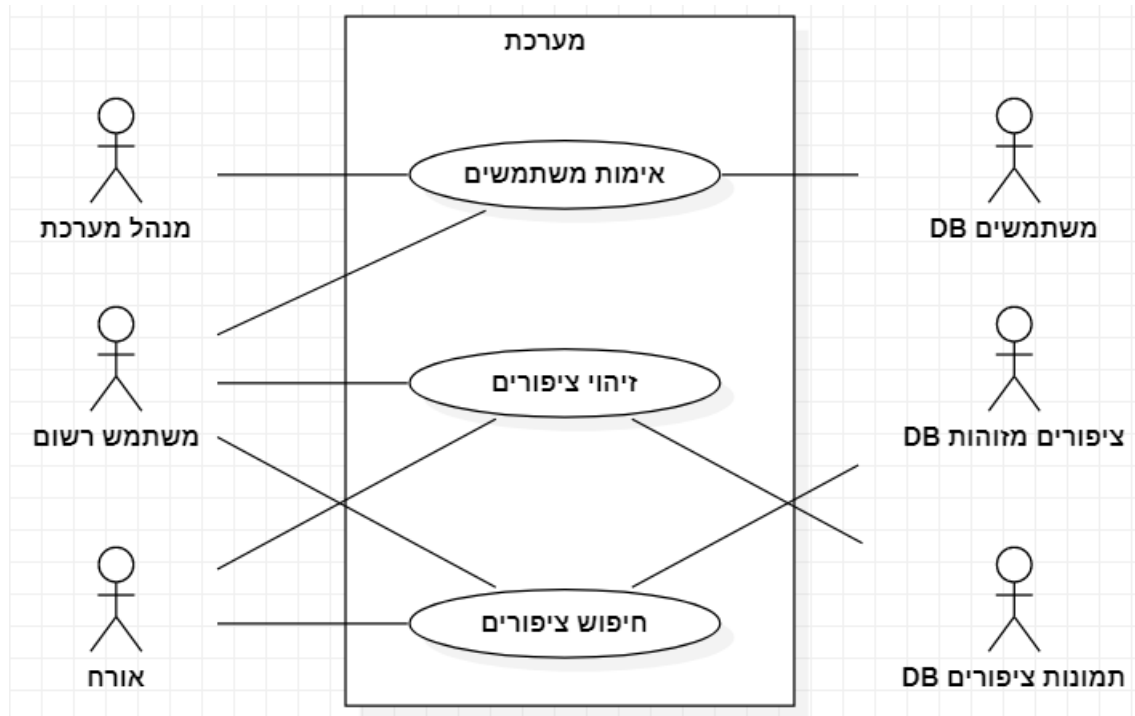
הרכיבים העיקריים בהם המערכת תשתמש יהיו מאגרי המידע החיצוניים, אלגוריתם לזיהוי הציפור מתמונה, אלגוריתם לחיפוש ציפורים באזור ובניית היישומון למשתמשים.

#### **8.5.2. האתגר ההנדסי של המערכת**

האתגר ההנדסי במערכת מהווה פיתוח אלגוריתם ראייה ממוחשבת זיהוי ציפור על ידי תמונה שצולם בטלפון נייד, המערכת צריכה לדעת לנקות את הדברים ה"לא קשורים" לתמונה לזהות את סוג הציפור אותה צילמו. המערכת תצטרך להתגבר על המכשול של צילום תמונה בעזרת מכשיר נייד מה שיכול להיות באיכות לא טובה או שאינה ממרכזת את החלק העיקרי של התמונה שזו בעצם הציפור שאותה המשתמש ירצה לזהות בעזרת האפליקציה.

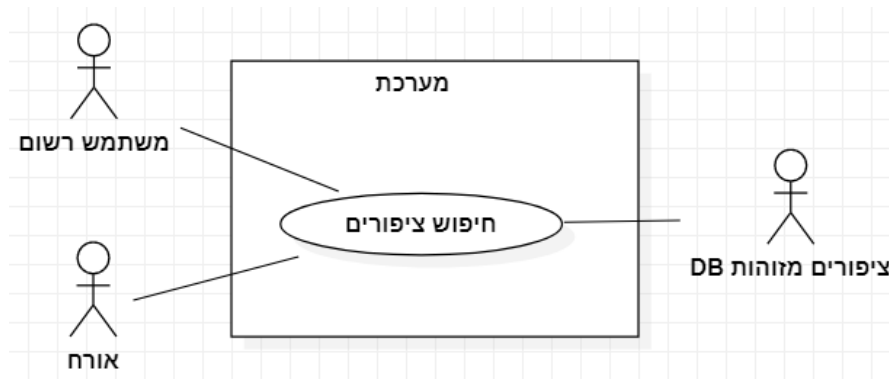
## 9. ניתוח פונקציונלי ראשוני

באיור 9.1 ניתן לראות את הניתוח הפונקציונלי הראשוני כחלק מתרשים.



איור 9.1 – תרשים Use-case

## 9.1. חיפוש ציפור



איור 9.1.1 – מקרה של חיפוש ציפורים

### 9.1.1. זרימה ראשית

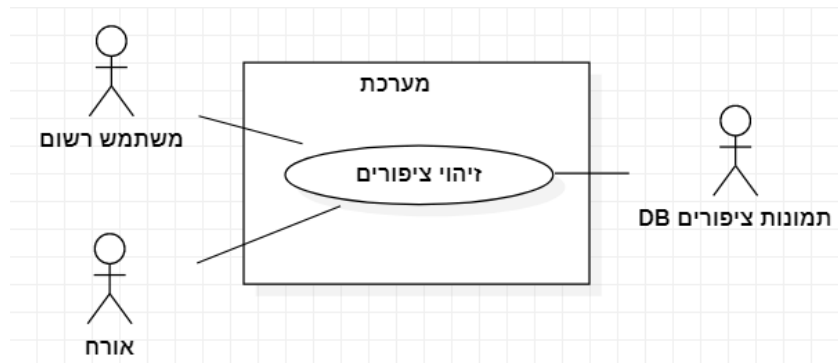
1. המשתמש נכנס למערכת (כאורח או כמשתמש)
2. המערכת שואלת את המשתמש שאלות על הציפור אותה מחפש לפי שם, לפי מיקום או לפי משפחת ציפורים.
3. המשתמש מזין את שם הציפור אותה מחפש, או עונה על לפחות אחת מהשאלות בבחירת המאפיינים.
4. המערכת מחפשת מתוך כלל הציפורים בDB את הציפורים העונות על אותן השאלות.
5. המערכת מציגה למשתמש את הציפורים העונות על אותן הקריטריונים שהציג.
6. המשתמש בוחר את הציפור המבוקשת.
7. המערכת מציגה מידע אודות אותה הציפור.

## 9.1.2. זרימה משנית א

תרשים זרימה משנית בעקבות סעיף 4 בזרימה הראשית

1. המערכת בודקת אם הוזן שם - במידה ולא הוזן, עוברים לשלב 6 בזרימה משנית זו.
2. המערכת מחפשת במאגר הציפורים ציפור בעלת אותו שם.
3. אם המערכת מוצאת ציפור בעלת אותו שם - המשך בזרימה הראשית בשלב 7.
4. אם לא נמצאה ציפור בעלת אותו השם - המערכת שולחת הודעה למשתמש שלא נמצאה ציפור בשם זה.
5. המשך לשלב 3 בזרימה הראשית.
6. המערכת בודקת אם נבחרו משפחות ציפורים - במידה ולא נבחרו, עוברים לשלב 9 בזרימה משנית זו.
7. המערכת מחפשת את כל הציפורים במערכת מאותן משפחות ציפורים.
8. אם המערכת מוצאת ציפורים מאותה המשפחה - המשך בזרימה הראשית בשלב 5.
9. המערכת בודקת מהו מיקום המשתמש ומה הרדיוס אותו בחר - במידה ולא הוזן, עוברים לשלב 15 בתרשים זה.
10. המערכת מחפשת את שמות הציפורים שנמצאו ברדיוס של אותו המיקום במערכת.
11. המערכת מחפשת במאגר המידע את הציפורים עם אותם השמות.
12. אם המערכת מוצאת ציפורים/אלו - המשך בזרימה הראשית בשלב 5.
13. במידה ולא נמצאו ציפורים ברדיוס זה - המערכת שולחת הודעה למשתמש שלא נמצאו ציפורים ברדיוס הנבחר במיקום הנוכחי.
14. המשך לשלב 3 בזרימה הראשית.
15. המערכת מתריעה שלא נבחרה דרך חיפוש.
16. המשך לשלב 3 בזרימה הראשית.

## 9.2. זיהוי ציפורים



איור 9.1.2 – מקרה של זיהוי ציפורים

### 9.2.1. זרימה ראשית

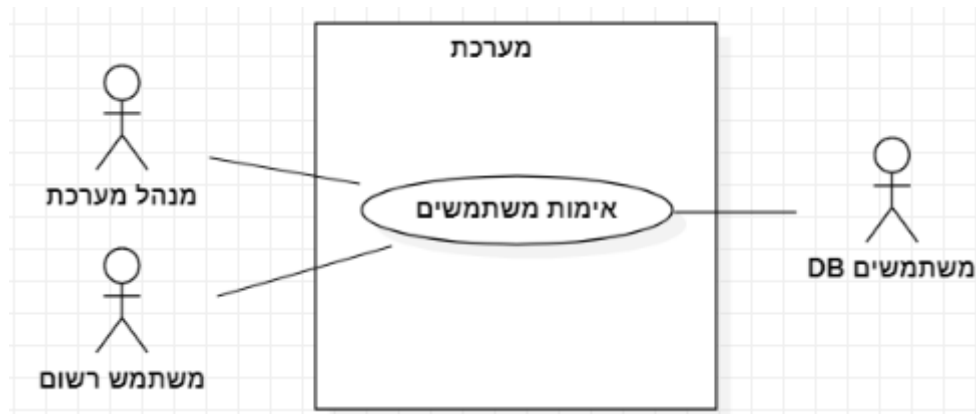
1. המשתמש נכנס למערכת ומבקש לזהות ציפור.
2. המערכת מבקשת מהמשתמש מידע על המיקום שלו ותמונה של הציפור.
3. המשתמש מאפשר קבלת מיקום ומעלה למערכת תמונה של הציפור.
4. התמונה נשלחת לזיהוי אצל האלגוריתם.
5. האלגוריתם מנקה את התמונה מרעשים.
6. האלגוריתם שולח את התמונה למודל המאומן לקבלת תוצאה.
7. האלגוריתם מחזיר תשובה.
8. המערכת שולחת את פרטי הזיהוי (תמונה, מיקום ונתונים) לאימות.
9. המערכת מחזירה למשתמש מידע אודות הציפור שזוהתה.

### 9.2.2. זרימה משנית א

זרימה משנית בעקבות סעיף 7 בזרימה הראשית:

1. האלגוריתם בודק אם התוצאה המיטבית של שם הציפור היא ברמת זיהוי של מעל 40% לאותה הציפור.
2. במידה וקיימת ציפור כזו, עבור לשלב 8 בתרשים הזרימה הראשי.
3. האלגוריתם בודק אם התוצאה המיטבית של משפחת הציפורים היא ברמת זיהוי של מעל 50% לאותה הציפור.
4. במידה וקיימת משפחת ציפורים כזו, החזר מידע אודות משפחת הציפורים שזוהתה ומידע על הציפורים באותה המשפחה.
5. במידה והמשתמש רשום (כצפר מקצועי) המערכת תאפשר למשתמש לבחור איזו ציפור מאותה משפחה מתאימה לזיהוי זה.
6. המשך לשלב 9 בתרשים הראשי.

### 9.3. אימות צפר מקצועי



איור 9.1.4 – מקרה של אימות צפר מקצועי

#### 9.3.1. זרימה ראשית

1. משתמש מציין בזמן הרישום שהוא צפר מקצועי.
2. המערכת מבקשת ממנו תמונה/ מסמך של תעודת הסמכה.
3. המשתמש מעלה למערכת.
4. המשתמש נרשם כמשתמש רגיל ורק לאחר אישור מנהל המערכת יסומן כצפר מקצועי.
5. מנהל המערכת נכנס לאישור משתמשים במערכת.
6. המערכת מציגה למנהל המערכת את פרטי הרישום ואת תעודת ההסמכה של הצפר.
7. מנהל המערכת מחליט אם לאשר או לדחות את הבקשה.
8. במידה ומנהל המערכת מאשר - המשתמש מסווג כצפר מקצועי.



## 10. הצגה ראשונית של חלופות למימוש

### 10.1. אלגוריתם לזיהוי ציפור

אלגוריתם הזיהוי מהווה חלק משמעותי ובלתי נפרד מהפיתוח של המערכת. ללא זיהוי נכון של הציפורים - המערכת תפספס את המטרה המרכזית שלה. לכן חשוב לבחור בחלופה המתאימה ביותר לזיהוי המבוקש.

#### 10.1.1. פיתוח באמצעות שימוש באלגוריתמים מבוססי תכונות התמונה

שימוש באלגוריתמים מבוססי שימוש מאפיינים בתוך התמונות עצמן – למשל שוני בצבעים, קווי מתאר, גדלים ויחסים וכו'. בדרך זו נוכל לזהות על ידי מאפיינים דומים ציפורים ומשפחות ציפורים.

#### 10.1.2. פיתוח באמצעות שימוש באלגוריתמים מבוססי ראייה ממוחשבת ולמידה עמוקה

שימוש באלגוריתמים מבוססי ראייה ממוחשבת קיימים, שמטרתם תהיה לזהות את הציפור או את משפחת הציפורים של הציפור בתמונה. נאמן את כלל האלגוריתמים על מאגר תמונות הציפורים, נבצע שינויים על התמונות (אוגמנטציות) כדוגמת חיתוך, הזזה, סיבוב, חידוד וטשטוש וכך נגדיל את מאגר האימון של האלגוריתם, נשווה בין התוצאות שהתקבלו בכל האלגוריתמים ונבחר בתוצאה המדויקת ביותר.

#### 10.1.3. פיתוח באמצעות שילוב האלגוריתמים

שילוב בין אלגוריתמים מבוססי תכונות התמונה ואלגוריתמים מבוססי ראייה ממוחשבת ולמידה עמוקה. בדרך זו נוכל לאמן בדרכים שונות את המאגר ולשלב את היתרונות שבשתי השיטות.

#### 10.1.4. השוואת החלופות

| שילוב האלגוריתמים  | אלגוריתמים מבוססי ראייה ממוחשבת ולמידה עמוקה   | אלגוריתמים מבוססי תכונות התמונה  |                         |
|--|--|--|-------------------------|
| חיבור למאגר מידע, ספריות OpenCV, התקנת Python על המחשב.  |  |  | <b>רכיבים רלוונטיים</b> |
| כתיבת אלגוריתם המתבסס גם על מודלים של תכונות תמונה וגם מודלים של למידה עמוקה.                  | כתיבת אלגוריתם המתבסס על מודלים של למידה עמוקה לצורך זיהוי התמונות.  | כתיבת אלגוריתם המתבסס על מודלים של תכונות תמונה בו אנו נחליט מהם המאפיינים הדומים                              | <b>אופן מימוש</b>       |
| מימוש מורכב  | מימוש מורכב.   | מימוש פחות מורכב.  | <b>רמת קושי למימוש</b>  |
| שילוב היתרונות שבאלגוריתמים מבוססי תכונות תמונה ובאלגוריתמים מבוססי ראייה ממוחשבת ולמידה עמוקה | התאמת האלגוריתם במדויק לצרכים שלנו, לפי הקריטריונים עליהם אנחנו רוצים להתבסס. במימוש והשוואה בין מספר מודלים התוצאה תהיה מדויקת יותר | התאמת האלגוריתם במדויק לצרכים שלנו, לפי הקריטריונים עליהם אנחנו רוצים להתבסס. מהיר יותר ממימוש של בלמידה עמוקה | <b>יתרונות</b>          |
| זמן אימון ארוך מכל שיטה בנפרד.   | דורש זמן אימון וזמן לקבלת תוצאה ארוך מזה של אלגוריתם מבוסס תכונות  | האלגוריתם מתבסס בצורה יותר שטחית מעמוקה בזיהוי הציפור בתמונה   | <b>חסרונות</b>          |
| 8  | 7  | 6  | <b>משקל התרומה</b>      |

טבלה 10.1 – השוואת חלופות אלגוריתם.

#### 10.1.5. החלופה הנבחרת

בהתאם לאפשרויות שהוצגו, בחרנו להשתמש בשילוב בין האלגוריתמים. בדרך זו, נוכל גם לקבל את היתרונות של שתי השיטות וגם להכיר יותר טוב את המודלים הקיימים, לממש ולשלב אותם בצורה הנכונה – וזו מטרה עצמית שהצבנו לעצמנו בפרויקט הגמר.

## 10.2. דרכי פיתוח האלגוריתם

### 10.2.1. פיתוח אלגוריתם באמצעות Python

השימוש חינוכי, בPython קיימות ספריות רבות למתכנתים שרוצים לפתח בסביבה זו אלגוריתמים של ראייה ממוחשבת, כדוגמת Pytorch, Keras. בנוסף, קיימות ספריות כמו OpenCV לצורך עריכת תמונה. ממשק נוח בצורת קוד בלבד. מפתחים רבים בוחרים לממש ולפתח קוד בשפה זו.

### 10.2.2. פיתוח אלגוריתם באמצעות Matlab

השימוש חינוכי לסטודנטים, הממשק משתמש נוח ובצורת קוד בלבד. נוח למפתחים שאינם רוצים או יודעים לפתח בעזרת Python.

### 10.2.3. פיתוח בסביבת R

שימוש חינוכי, ממשק נוח בצורת קוד לבד. קיימות מגוון של ספריות לפיתוח האלגוריתם בסביבה זו. סטודנטים ובעלי מקצוע העוסקים בסטטיסטיקה וכלכלה מרבים להשתמש בה.

### 10.2.4. פיתוח אלגוריתם באמצעות C++

השימוש בשפה זו חינוכי, הפיתוח נעשה על ידי כתיבת קוד. בשפה זו ישנן ספריות רבות למתכנתים לבניית אלגוריתמי ראייה ממוחשבת, וכן גם ספריות כדוגמת OpenCV לעריכת התמונות.

מפתחים רבים בוחרים להשתמש בה כיוון ששפה זו מהירה משפות עיליות אחרות, אך הכתיבה בשפה זו פחות אינטואיטיבית משפות אחרות (כדוגמת Python).

### 10.2.5. השוואת החלופות

| C++                | R                  | Matlab  | Python           |                           |
|--------------------|--------------------|---|------------------|---------------------------|
| קוד בלבד           | קוד בלבד           | קוד בלבד  | קוד בלבד         | <b>ממשק משתמש</b>         |
| חינמי              | חינמי              | חינמי לסטודנטים בלבד, לאחר מכן בתשלום                 | חינמי            | <b>מחיר</b>               |
| V                  | V                  | יש לכתוב את הקוד בPython או בC++ ולהמיר לקובץ Matlab. | V                | <b>תמיכה בOpenCV</b>      |
| Visual Studio Code | R                  | Matlab  | Jupyter Notebook | <b>סביבת פיתוח אפשרית</b> |
| ממוצעת             | נוח                | נוחות בינונית   | נוח מאוד         | <b>נוחות לכתיבת קוד</b>   |
| V                  | באופן חלקי         | V   | V                | <b>תמיכת הסביבה בGPU</b>  |
| בינוני למימוש      | קל - בינוני למימוש | קל - בינוני למימוש                                    | קל למימוש        | <b>רמת קושי למימוש</b>    |

טבלה 10.2 – השוואת חלופות דרכי פיתוח האלגוריתם.

### 10.2.6. החלופה הנבחרת

בהתאם לאפשרויות שהוצגו, בחרנו להשתמש ב-Python, כיוון שהשימוש בשפה זו עבור פיתוח אלגוריתמי ראייה ממוחשבת נפוצה ורלוונטית. בנוסף, השפה נוחה עבורנו לשימוש (גם לאחר שהתנסינו בה), תומכת בOpenCV ובימוש בGPU.

## 10.3. מימוש פלטפורמת SDK

### 10.3.1. פיתוח באמצעות Native Android

המשמעות בשימוש בפלטפורמה זו היא פיתוח יישומון תואם ספציפית לפלאפונים בעלי מערכת הפעלה מסוג Android. חברות רבות כדוגמת Samsung, LG, OnePlus משתמשות במערכת הפעלה זו, ולכן מערכת הפעלה זו מהווה נתח גדול משוק המכשירים הניידים החכמים.

מערכת הפעלה זו תומכת בשיטת הקוד הפתוח ושפות התכנות לפיתוח ב-Native Android אלו Java, Kotlin.

במידה ונבחר לממש את האפליקציה בפלטפורמת Native Android, נוכל לאפשר למכשירים בעלי מערכת הפעלה של Android בגרסאות שונות להריץ את היישומון. אך שימוש בשיטה זו ימנע מאיתנו פיתוח למכשירים בעלי מערכת הפעלה IOS.

### 10.3.2. פיתוח באמצעות Native IOS

המשמעות בשימוש בפלטפורמה זו היא פיתוח יישומון תואם ספציפית לפלאפונים בעלי מערכת הפעלה מסוג IOS. החברה היחידה התומכת במערכת הפעלה זו היא Apple, אשר בעצמה מהווה נתח מכובד משוק המכשירים הניידים החכמים.

בכדי לפתח בסביבה זו, יש צורך להשתמש במחשב נייד של חברת Apple בלבד, וגם להשתמש בתוכנה xCode לפיתוח היישומון. השפות בהן ניתן לפתח ב-Native IOS אלו Objective C, Swift.

במידה ונבחר להשתמש בפלטפורמה זו, הדבר ימנע מאיתנו פיתוח למכשירים בעלי מערכת הפעלה Android.

### 10.3.3. פיתוח בפלטפורמת React Native

פלטפורמת React Native נכתבת ב-Java Script ופותחה ע"י חברת Facebook. פלטפורמה זו אינטואיטיבית לבניית ממשק משתמש עבור מפתחים ומאפשרת מגוון אפשרויות פיתוח.

פלטפורמה זו מאפשרת לנו לפתח במקביל יישומון למכשירים חכמים בעלי מערכת הפעלה Android וגם למכשירים חכמים בעלי מערכת הפעלה IOS. דהיינו, פיתוח במקביל לשת מערכות ההפעלה, מה שמקצר בהרבה את שלבי הפיתוח לעומת פיתוח יישומון נפרד לכל מערכת הפעלה.

### 10.3.4. פיתוח בפלטפורמת Flutter

אפשרות זו גם כן פותח את האפשרות לפתח במקביל למכשירים בעלי מערכת הפעלה Android וגם למכשירים בעלי מערכת הפעלה IOS ביחד. באותו פיתוח כותבים לשניהם, מה שמקצר הרבה את שלבי הפיתוח לעומת פיתוח שני יישומונים שונים לכל מערכת הפעלה. Flutter נכתב בשפת Dart ופותחה על ידי חברת Google.

### 10.3.5. השוואת החלופות

| Flutter  | React Native  | Native IOS                 | Native Android                          |                       |
|--|---|----------------------------|---|-----------------------|
| Visual Studio Code   | Visual Studio Code, Android Studio (emulator)   | מחשב Mac סביבת פיתוח Xcode | Android Studio                          | סביבת פיתוח           |
| v  | V   | X                          | V                                       | תמיכה במכשירי Android |
| V  | V   | V                          | X                                       | תמיכה במכשירי IOS     |
| פיתוח לשני מערכות ההפעלה Android, IOS באותו הפרויקט, תמיכה של Google | פיתוח לשתי מערכות הפעלה Android, IOS באותו הפרויקט, תמיכה של Facebook, אפשרויות עיצוב מגוונות | תאימות למכשירי IOS.        | תאימות למכשירי Android, סביבת קוד פתוחה | יתרונות               |
| Dart   | Java Script   | Objective C, Swift         | Java, Kotlin                            | שפות תכנות            |
| מימוש פשוט   | מימוש פשוט  | מימוש מורכב ויקר           | מימוש פשוט                              | רמת קושי למימוש       |

טבלה 10.3 – השוואת חלופות מימוש פיתוח SDK.

### 10.3.6. החלופה הנבחרת

בהתאם לאפשרויות שהוצגו, בחרנו בחרנו בפיתוח ב- React Native כיוון שJavaScript מוכר לנו (לעומת Dart ו-Swift למשל) ובנוסף, נוכל לפתח עבור IOS ו-Android יחדיו.

## 11. סקר שוק ראשוני

### 11.1. יישומון Merlin Bird ID [5]

Merlin Bird ID הינו יישומון שמטרתו היא לזהות ציפורים לפי תמונה וקול. האלגוריתם של היישומון פותח ע"י Cornell Labs. היישומון זמין למכשירי Android ומכשירי IOS היישומון הינו חינמי. כדי להשתמש ביישומון זה המשתמש צריך להוריד את היישומון מ Google Play למכשירי Android ו Apple Store למכשירי IOS. לאחר הורדת היישומון, יש להירשם ואז אפשר להתחיל להקליט ציפורים או לצלם ציפורים ולקבל זיהוי של הציפורים לפי האלגוריתם של היישומון.

היישומון מתאים לכל הגילאים ומאוד קל לשימוש לכל המשתמשים בכל הגילאים. בנוסף כאשר האפליקציה מבצעת זיהוי של ציפור היא מביאה גם נתונים עליה.

#### 11.1.1. יתרונות:

- מבצע זיהוי של ציפור גם לפי תמונה וגם לפי קול, ישנם שתי דרכים לבצע זיהוי של הציפור. כך שמשתמש יכול לבצע את הזיהוי בכמה דרכים.
- היישומון צבר לאורך השנים מוניטין והוא מאוד פופולארי בקרב חובבי צפרות.

#### 11.1.2. חסרונות:

- נכון להיום אי אפשרות לחפש ציפורים לפי מיקום גאוגרפי ספציפי (אלא ע"י מדינה).
- נכון להיום לא ניתן למצוא מקומות קודמים שציפור מסוימת נראתה לאחרונה.

### 11.2. אתר BirdNET [6]

אתר BirdNET הינו פלטפורמה מחקרית- אזרחית, המאפשרת לזהות ציפורים ע"י צלילים. האלגוריתם לזיהוי מבוסס על למידת מכונה ורשת עצבית מלאכותית ופותח ע"י Cornell Labs. שתי הדרכים המרכזיות להשתמש בפלטפורמה זו היא ע"י העלאת קובץ שמע לאתר, באתר מתבצע הזיהוי ומוצגות תוצאות אפשריות לציפורים המשמיעות את צליל זה, או - ע"י הורדת היישומון, רישום והעלאת קבצי שמע/ הקלטה דרך המכשיר החכם.

#### 11.2.1. יתרונות:

- יוצרי הפלטפורמה מעדכנים באופן תדיר את האלגוריתם ואת מספר הציפורים שמזהה.
- ממשק קל לשימוש

#### 11.2.2. חסרונות:

- מזהה לפי שמע בלבד.
- הציפורים המוכרות לאפליקציה מאזור צפון אמריקה ואירופה.
- באתר הפלטפורמה, ניתן לקבל רק את שם הציפור ולא מידע עליה

### 11.3. אתר הצפרות הישראלי [7]

אתר הצפרות הישראלי שייך לחברה להגנת הטבע והוקם בסיוע של ציוני דרך, המחלקה לזואולוגיה באוניברסיטת תל אביב, קרן הדוכיפת ומשרד התיירות. בתוך אתר זה ניתן למצוא מידע על מרכזי צפרות בארץ, פעילויות, מאמרים, תצפיות וטיולים. בנוסף לכל אלו, באתר זה קיים מגדיר ציפורים. המגדיר הינו מוקד הידע המקיף ביותר על כלל ציפורי ישראל. בשביל למצוא ציפור ניתן לחפש אותה במגדיר ולקבל מידע על הביולוגיה שלה, זיהוי, תפוצה, תצפיות אחרונות ותמונות.

#### 11.3.1. יתרונות:

- מציג מידע כולל ומקיף אודות הציפורים בישראל.
- מונגש בעברית.
- מאפשר לראות מיקומים של אותן הציפורים בארץ.

#### 11.3.2. חסרונות:

- לא ניתן לזהות ציפור לפי תמונה או שמע.
- לא ניתן לחפש ציפורים לפי אזורים.

### 11.4. לקסיקון מפה: הציפורים - המדריך השלם לציפורי אירופה וישראל [8]

מגדיר ציפורים זה הינו ספר הכולל מידע מרוכז על מאות רבות של ציפורים ומציג מידע אודותן ואיורי צבע עליהן. הספר מכיל את כל המידע הדרוש לזיהוי ציפורים בכל עונות השנה, תיאורי מינים וציפורים, מקום החיות שלה, תפוצה, מאפיינים, מפות וכו'. הספר הינו מפעל משותף של הוצאת מפה, הוצאת הקיבוץ המאוחד ובשיתוף עם החברה להגנת הטבע, מרכז הצפרות הישראלי והמרכז הבינלאומי לחקר נדידת הציפורים בלטרון.

#### 11.4.1. יתרונות:

- אינו מצריך חיבור לרשת
- מונגש בעברית.

#### 11.4.2. חסרונות:

- הספר מכיל מעל 400 עמודים ובכדי לחפש ציפור צריך לחפש אותה לפי בצורה ידנית.
- לא ניתן לזהות ציפור בצורה "חכמה" לפי תמונה שמעלים, אלא לחפש מבין האיורים בצורה ידנית מהו האיור המתאים ביותר לאותה ציפור אותה אתה רוצה לזהות מהתמונה.



## 11.5. השוואה

| אתר הישראלי                           | אתר הצפרות                                   | לקסיקון מפה: הציפורים | יישומון Merlin Bird ID                       | אתר birdnet                | יישומון שלנו                              |
|---------------------------------------|--|-----------------------|--|----------------------------|---|
| לא                                    | לא   | לא                    | כן   | לא                         | כן  |
| כן                                    | כן   | לא                    | כן   | כן                         | כן  |
| כן                                    | כן   | לא                    | לא   | לא                         | כן  |
| כן                                    | כן   | לא                    | לא ידוע                                      | לא ידוע                    | לפי תמונות שהועלו ע"י מנהל מערכת ומשתמשים |
| לא                                    | לא   | לא                    | בחירת אזור לפי מדינה, לא מציג מיקום מדויק.   | לא                         | בחירת אזור לפי רדיוס                      |
| מחשב, מכשיר חכם                       | ספר מודפס                                    | מכשיר חכם             | מחשב, מכשיר חכם, בקרי Arduino ו-Raspberry pi | מכשיר חכם                  | שימוש באמצעות                             |
| ציפורים בישראל / נודדות מעל שמי ישראל | ציפורים בישראל עד נקודת הזמן בה המגדיר הודפס | עולמי                 | צפון אמריקה, אירופה                          | ציפורים שעוברות בשמי ישראל | ריכוז הציפורים                            |
| עברית, אנגלית                         | עברית  | אנגלית                | אנגלית                                       | אנגלית                     | שפת נגישות                                |
| חינם, ניתן לתרום                      | 169 ש"ח                                      | חינם                  | חינם, ניתן לתרום                             | חינם                       | עלויות                                    |

טבלה 11.1 – השוואת סקר שוק.

## 12. האמצעים/ הכלים הנדרשים

לתהליך פיתוח אפליקציית הלקוח, ניעזר באימולטור המובנה במחשב (Android Studio) המדמה הפעלת מכשיר Android באופן ממוחשב, בנוסף נשתמש במכשירי Android בבעלותנו לביצוע בדיקות נוספות מחוץ לאימולטור.

לביצוע האלגוריתם נשתמש בשפת Python, בה קיימות ספריות מוכנות לראייה ממוחשבת ולרשתות נוירונים. השפה Python מאפשרת לנו ביעילות לבצע עריכה לתמונות בראייה ממוחשבת, בניית אלגוריתם לזיהוי הציפור והצגת סטטיסטיקה בזמן אימון האלגוריתם.

את צד השרת גם כן נכתוב בשפת Python וזאת משום שהאלגוריתם שלנו כתוב בשפת Python נרצה לממש את המקסימום מאלגוריתם זה, בכל שנשאר באותה שפה כמו האלגוריתם נוכל בקלות להתממשק איתו בצד השרת, ששם נרצה גם להשתמש בו וגם לאמן אותו לאורך תקופות.

### 12.1. התייחסות לסביבת הפיתוח וסביבת ההרצה

סביבות הפיתוח בהן נשתמש יהיה סביבות תומכות באימולטורים של Android כך שיהיה ניתן לבדוק את צד המשתמש בתהליך הפיתוח באופן נוח בעזרת המחשב ללא תלות בחומרה נוספת. סביבת הריצה תהיה באופן דומה בפלטפורמת Android.

## 13. תוצרי הפרויקט

### 13.1. תיאור תוצרי הפרויקט

#### 13.1.1. אלגוריתם לזיהוי ציפור לפי תמונה

פיתוח אלגוריתם המקבל תמונה של ציפור, מנתח את התמונה לחלקים המתאימים בתמונה וחיזוי של סוג הציפור אותה האלגוריתם קיבל. האלגוריתם יקבל את התמונה, יחלק את ניתוח המאפיינים של כל חלק בציפור ולאחר מכן יבצע חיזוי של הציפור בתמונה אותה קיבל. פיתוח האלגוריתם יעשה באמצעות מימוש של מודלים של ראייה ממוחשבת ורשתות נוירונים.

#### 13.1.2. אלגוריתם חיפוש ציפורים לפי אזור

אלגוריתם המקבל את המיקום הרצוי, רדיוס, וזמן לבדוק לאחר. האלגוריתם מציג את כל הציפורים שנחזו באזור תחום זה המורכב בעזרת מיקום מרכזי ורדיוס ומשם מסנן את הציפורים שנראשו באזור זמן זה.

#### 13.1.3. פיתוח יישומן למכשיר נייד לזיהוי וחיפוש הציפורים

פיתוח אפליקציה למכשירים סלולריים חכמים לשימוש של חובבי צפרות או אנשים המתעניינים באותו הרגע לזיהוי הציפור. מטרת היישומן היא שמשתמש יכול לקחת את המכשיר הנייד שלו ולזהות באותו הרגע ציפור שהוא נתקל בה, או לחפש ציפור המעניינת אותו לראות. המשתמש יכול לחפש אותה לפי היכן נראתה לאחרונה. בנוסף אם המשתמש רצה לחפש את הציפורים הנמצאות באזורי, הוא יכול לעשות זאת בעזרת שליחת המיקום שלו ורדיוס שהוא רוצה למצוא ציפורים, וכך הוא יקבל את הציפורים שנראו באזור בזמן האחרון.

### 13.2. תיאור התוצרים בהתייחס למערכת הסופית

היישומן (13.1.3) יאפשר לבצע את הפעולות זיהוי לפי תמונה - בעזרת האלגוריתם המפורט (13.1.1) וחיפוש ציפור לפי שם או מיקום - בעזרת האלגוריתם מסעיף (13.1.2). המערכת תשלב אותם בתוך היישומן לצורך קבלת זיהוי מהיר עבור המשתמש.

### 13.3. שלבי הפיתוח והבדיקות

פיתוח ראשוני של האלגוריתם לזיהוי לפי תמונה יתבצע במקביל ללימודי הספרייה והתנסות בפיתוח היישומן עצמו. לאחר מכן, המשך שיפור האלגוריתם לזיהוי התמונה הראשוני, אימונו, הוספת נתונים וציפורים אליו תתבצע במקביל לפיתוח היישומן עצמו. לאחר סיום האלגוריתם לזיהוי התמונה והגעה לתוצאה מספקת, נפתח את אלגוריתם החיפוש לפי שם או מיקום.

הבדיקה של האלגוריתם לזיהוי התמונה יתבצע לפי קבוצות אימון וולידציה. בנוסף, לצורך אימון שלישי (test) נשתמש בתמונות שאינן היו במאגר בכלל ולא נכללו כחלק מקבוצת וולידציה או אימון ונבדוק את התוצאות לפיהן.

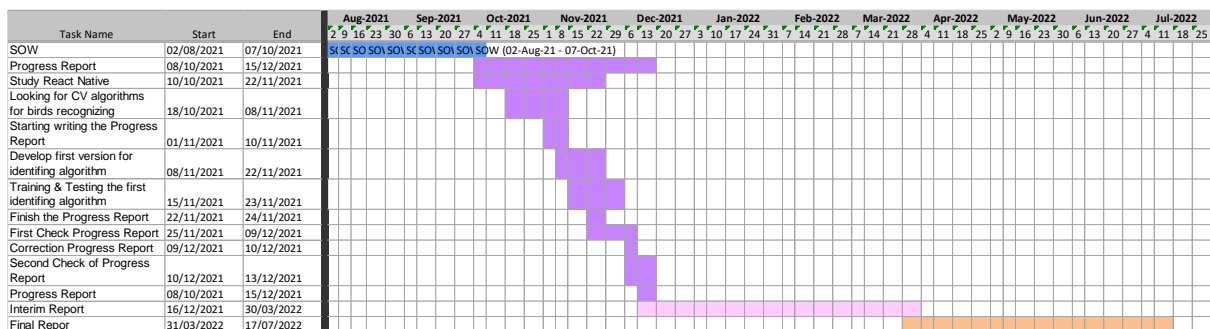
הבדיקות על אלגוריתם החיפוש יכללו בדיקות על אופן השימוש באלגוריתם ונבצע באפליקציה בדיקות סימולציה עם נתוני סרק ועם מקרי קיצון שיש לטפל בהם (למשל - שם של ציפור שאינה מוכרת למערכת, אזור ריק של ציפורים וכו').

לבסוף נבצע ביישומן עצמו בדיקות אמיתיות (hands on) על טיב היישומן ועמידותו במדדים מול מקרי קיצון אמיתיים ובכך לבדוק את יעילותו בזמן ההרצה.

## 14. תוכנית עבודה ראשונית לביצוע הפרויקט

| Task Name       | Start      | End        |
|-----------------|------------|------------|
| SOW             | 02/08/2021 | 07/10/2021 |
| Progress Report | 08/10/2021 | 15/12/2021 |
| Interim Report  | 16/12/2021 | 30/03/2022 |
| Final Repor     | 31/03/2022 | 17/07/2022 |

טבלה 14.1 – תאריכי הגשות לביצוע הפרויקט.



טבלה 14.2 – GANTT הגשות לביצוע הפרויקט.

## 15. פערים

### 15.1. פערי ידע

נכון לתאריך הגשת ה-SOW, את תנאי הקדם לפרויקט הגמר (הקורס מבוא לראייה ממוחשבת) עברנו במהלך התואר ואין קורסים נוספים בתואר שאנו מחויבים לעבור עבור הפרויקט.

מבחינת פיתוח התוכנה בצד הלקוח, עלינו ללמוד את הספרייה בה אנו בחרנו לבצע את פיתוח היישומון, React Native.

מבחינת פיתוח התוכנה בצד השרת, עלינו ללמוד את הספריות לכתיבת צד השרת בשפת Python.

מבחינת פיתוח האלגוריתם לזיהוי ציפורים, אין פערים בנושא וניתן להתחיל לפתח את האלגוריתם.

### 15.2. זמינות הציוד הנדרש

לצורך פיתוח האלגוריתם והרצה שלו, נזדקק לשטח אחסון בענן בשביל לשרת בצורה מרוחקת את משתמשי היישומון בעת זיהוי התמונות, וגם לצורך שמירת מאגר התמונות של האימון, מאגר המשתמשים ומאגר הזיהויים.

הציוד הנדרש לפיתוח היישומון זה קיים על המחשבים האישיים שלנו ואין צורך בחומרה נוספת או רכיבים נוספים. הפיתוח יעשה בעזרת סביבות פיתוח חנמיות שאינן דורשות רישיונות.

## 16. סיכונים עיקריים ודרכי התמודדות

בעת ניתוח הסיכונים ניתן לכל סיכון ציון בין 1-5, כאשר 1 הכי נמוך ו-5 הכי גבוה.

אנו נביט על שלושה קריטריונים: סבירות שהסיכון יתממש, מידת ההשפעתו על הפרויקט, ומכפלה של ציוני שני הקריטריונים שתציין את רמת הסיכון בכדי לאמת אילו סיכונים ניתן לקבל ואילו גדולים מדי. כך נוכל לבנות תוכנית עבודה יותר מדויקת ולקבל שיקולים בהתנהלות כללית מול חלקי הפיתוח השונים או אפילו שינוי מלא של הפיצ'ר וכן נתייחס לדרכי המניעה האפשריים.

| # | הסיכון  | סבירות | מידת ההשפעה | רמת סיכון |
|---|---|--------|-------------|-----------|
| 1 | זיהוי לא נכון של ציפור קיימת באלגוריתם              | 4      | 5           | 20        |
| 2 | בעיית התממשקות עם Databases                         | 3      | 5           | 15        |
| 3 | אי התאמת המערכת לכלל המכשירים החכמים                | 3      | 3           | 9         |
| 4 | אי סנכרון מפות ותוצאות אלגוריתמים בין משתמשים שונים | 3      | 3           | 9         |
| 5 | בעיות התממשקות עם מערכת מפות חיצונית                | 2      | 4           | 8         |
| 6 | חלוקת עבודה לא יעילה                                | 2      | 3           | 6         |
| 7 | ריצה איטית של האלגוריתם                             | 3      | 2           | 6         |
| 8 | אי מציאת ציפור בחיפוש                               | 3      | 1           | 3         |

טבלה 16.1 – טבלת סיכונים עיקריים ודרכי התמודדות.

### 16.1. זיהוי לא נכון של ציפור קיימת באלגוריתם (20)

זיהוי הציפורים הוא החלק המרכזי במערכת. בלעדיו כמובן, לא ניתן לזהות את הציפורים. פיתוח ואימון של אלגוריתם זה יכול להיות ממושך ולמנוע מאיתנו לממש את מטרות המערכת בלוח זמנים שהצבנו לעצמנו.

#### 16.1.1. דרכי מניעה

יש לאמן את האלגוריתם עד שיגיע לתוצאת אימון רצויה ולהמשיך לאמן אותו כאשר המערכת מקבלת נתונים חדשים. בנוסף, נתבונן במאמרים ובמאגרי מידע על נושא זיהוי עצמים בתמונות ובבדוק מספר מודלים לאימון המערכת לפני שנגיע להכרעה מהי דרך האימון הטובה ביותר.

### 16.2. בעיית התממשקות עם מסד נתוני המערכת (15)

בעיית התממשקות עם מסד הנתונים של המערכת יכול לגרום לאיבוד מידע חשוב למערכת שלנו (משתמשים רשומים, מאגר מידע של ציפורים, מאגר הציפורים של האפליקציה וכו'). דבר זה פוגע באמינות היישומן, בהשגת מטרותיו העיקריות. שחזור מידע שאבד או נפגם יכול לגרום זמן רב שיפגע בזמני הפיתוח של המערכת. אי התממשקות עם Databases יכול לקחת זמן רב מהמערכת שלנו ובכך לעקב פיתוחים עתידיים במערכת במסגרת זמן הפרויקט.

#### 16.2.1. דרכי מניעה

נבחר מראש את סוג Databases האופטימלי עבורנו, ונוודא אפשרות התממשקות עם מסד נתונים זה ע"י השפה בה נשתמש וסביבת הפיתוח שבחרנו. בנוסף, נגבה את המידע הקיים באותו מסד נתונים מקומית או בענן אחר - כך שאם יקרה משהו למסד נתונים בו נשתמש, יהיה לנו גיבוי של המידע.

### **16.3. אי התאמת המערכת לכלל המכשירים החכמים (9)**

במידה ועיצוב היישומון לא יידע להתאים את עצמו לכלל המכשירים החכמים, חלק מהמכשירים לא יאפשרו למשתמש להשתמש ביישומון בצורה מיטבית.

אם זה מבחינה עיצובית, למשל שחלק מהטקסטים יחתכו או האפליקציה לא תממש את הפוטנציאל הוויזואלי שלה. מצב זה יכול לנבוע בעקבות שוני ברזולוציות מכשירים שונים. ואם זה מבחינה פונקציונלית, למשל שקיימת אפשרות ששוני בין גרסאות שונות של מכשירים / מערכות הפעולה תגרום גם לאיבוד של פונקציות שונות ביישומון.

#### **16.3.1. דרכי מניעה**

יש לתכנן את המסכים כך שידעו להתאים את עצמם לכל סוגי המכשירים הניידים, בנוסף - נוודא תאימות לאחור של הגרסאות.

### **16.4. אי סנכרון בין מפות ותוצאות אלגוריתמים של משתמשים שונים (9)**

כיוון שישנם משתמשים שונים ולכל משתמש מכשיר חכם אישי (ולא כולם משתמשים באותו המכשיר), במידה והאלגוריתם מתאמן מקומית, פרט לעומס שיכול להיווצר על המכשיר, מספר התוצאות בכל מכשיר שונה (ע"פ האימון והנתונים החדשים שהוא מקבל) גם תוצאת האימון יכולה להשתנות בין כל אחד ואחד מהמשתמשים (מה שיוכל להשפיע גם על תוצאות האלגוריתם לאותה תמונה במכשירים שונים) וגם, אם החיבור לרשת לא קיים יכול להיווצר מצב שמשתמשים לא יקבלו מיקומי ציפורים שניתנו ע"י משתמשים אחרים.

#### **16.4.1. דרכי מניעה**

את אימון האלגוריתם ועיבוד הנתונים נעשה על הענן, כך שהאלגוריתם יהיה מעודכן ומסונכרן עבור כלל המכשירים (הגישה אליו לא מקומית). בנוסף, מיקומי הציפורים שהועלו ע"י משתמשים ויופיעו על המסך, יתעדכנו במסד נתונים בכל חיבור לרשת - כך שכל טעינת מיקומים - המשתמש יכול לצפות במיקומים של כולם ולא רק באלו שלו.

### **16.5. בעיות התממשקות עם מערכת מפות חיצונית (8)**

התממשקות עם מערכות מפות כך שתציג את הנתונים על המפה זה דבר מאוד קריטי ביישומון. אנו צריכים למצוא מערכת זו להתממשק עם היישומון שלנו ותציג בתוכו את כל רצונותינו, זה יכול לפגוע דרישתנו להצגת ציפורים באזור אשר אנו רוצים לחפש.

#### **16.5.1. דרכי מניעה**

בשלב התכנון של פונקציה המציגה את מיקומי הציפורים על המפה, נוודא שהפלטפורמה להצגת מפה אותה נבחר ועליה נציג את מיקומי הציפורים במערכת מתאימה לדרישות שלנו.

## **16.6. חלוקת עבודה לא יעילה (6)**

במידה ולא נחלק את העבודה בצורה נכונה, יכול להיווצר מצב של איבוד זמן מיותר וניצול לא נכון של הזמן ושל המשאבים הקיימים. חלוקת עבודה נכונה בין הצוות היא חשובה מאד עבור הפיתוח של האלגוריתם וישומון בפרט, ושל פיתוח נכון של הפרויקט בכלל.

### **16.6.1. דרכי מניעה**

נתחיל על ידי תכנון המערכת וחלוקת העבודה לזמנים, יעדים, משימות עיקריות ותתי משימות. את משימות ותתי המשימות הקיימות בפרויקט נחלק לפי החזקות, הידע הקיים הרלוונטי ותחומי העניין של כל אחד מחברי הפרויקט. במהלך ביצוע הפרויקט נעבוד עם לוח הזמנים והיעדים אותו תכננו בהתחלה ונשתמש כדי לבחון את העמידה שלנו בזמנים שהצבנו לעצמנו בפרויקט. בנוסף, נשתמש בתוכנה לניהול משימות כדי לשמור במקום אחד את מצבי כל המשימות הקיימות בפרויקט כך שבנוסף לשיחות עדכון שוטפות, כל אחד מחברי הצוות יכול לדעת מה מצב כל המשימות.

## **16.7. ריצה איטית של האלגוריתם (6)**

במידה והחיבור לרשת יהיה איטי, תוצאות האלגוריתם יהיו יותר איטיות, כיוון התמונה תשלח מהמשתמש לאלגוריתם לאט יותר והתשובה מהאלגוריתם תתקבל למשתמש בצורה איטית יותר. בנוסף, יש לתכנן את האלגוריתם שידע לעבוד ביעילות ולא יקח לו הרבה זמן לנתח את התמונה.

### **16.7.1. דרכי מניעה**

יש לוודא כי היישומון מחובר לרשת. האלגוריתם ידע לבצע שינויים על התמונה (כדוגמת אוגמנטציות, חיתוכים ושינויי רזולוציה) כדי להעביר אותה אל שלב הזיהוי, וכך הזיהוי יתבצע בצורה מהירה יותר.

## **16.8. אי מציאת ציפור בחיפוש (3)**

כאשר משתמש מחפש ציפור ולא מוצא אותה בשורת החיפוש, אומנם המערכת תמשיך לפעול - אך המשתמש יחוה הרגשה של חוסר מידע מצד האפליקציה. אי מציאת הציפור יכולה להיגרם עקב כתיבת שם שגוי, אי חיבור לרשת או ציפור שאינה מוכרת למערכת.

### **16.8.1. דרכי מניעה**

יש לעדכן את הציפורים במערכת. במידה והציפור לא קיימת - נשקול להוסיף אותה למאגר הציפורים. במידה וקיימת במאגר, נבדוק אם החיבור לרשת קיים (מה שיכול למנוע מנתונים להגיע למשתמש). בנוסף, בשביל למנוע מצב בו שם הציפור שגוי, כאשר המשתמש מקליד את שם הציפור, המערכת תציע לו שמות של ציפורים בעלות שם דומה. במידה ולא נמצאה הציפור, המערכת תודיע למשתמש כי היא אינה מכירה ציפור עם אותו השם.



## 17. רשימת מקורות (ראשונית)

[1]<http://cs229.stanford.edu/proj2014/Aditya%20Bhandari,%20Ameya%20Joshi,%20Rohit%20Patki,%20Bird%20Species%20Identification%20from%20an%20Image.pdf>

[2]<https://www.microsoft.com/en-us/research/wp-content/uploads/2017/07/Look-Closer-to-See-Better-Recurrent-Attention-Convolutional-Neural-Network-for-Fine-grained-Image-Recognition.pdf>

[3]<https://ksiresearch.org/seke/seke20paper/paper122.pdf>

[4][https://www.researchgate.net/profile/Vani-Rajan/publication/339586575\\_CNN\\_BASED\\_CLASSIFIER\\_FOR\\_IDENTIFICATION\\_OF\\_CANINE\\_BREEDS/links/5e5a263892851cefa1cd98f2/CNN-BASED-CLASSIFIER-FOR-IDENTIFICATION-OF-CANINE-BREEDS.pdf](https://www.researchgate.net/profile/Vani-Rajan/publication/339586575_CNN_BASED_CLASSIFIER_FOR_IDENTIFICATION_OF_CANINE_BREEDS/links/5e5a263892851cefa1cd98f2/CNN-BASED-CLASSIFIER-FOR-IDENTIFICATION-OF-CANINE-BREEDS.pdf)

[5]<https://merlin.allaboutbirds.org>

[6]<https://birdnet.cornell.edu/>

[7]<https://www.birds.org.il/he>

[8][https://books.map.co.il/index.php?option=com\\_content&view=article&id=126:2011-08-24-09-41-24&catid=19&Itemid=15](https://books.map.co.il/index.php?option=com_content&view=article&id=126:2011-08-24-09-41-24&catid=19&Itemid=15)

[1] - Aditya Bhandari, Ameya Joshi, Rohit Patki . Bird Species Identification from an Image. Department of Computer Science, Stanford University, Department of Electrical Engineering, Stanford University, Institute of Computational Mathematics and Engineering, Stanford University

[2] - Jianlong Fu , Heliang Zheng , Tao Mei. Look Closer to See Better: Recurrent Attention Convolutional Neural Network for Fine-grained Image Recognition. Microsoft Research, Beijing, China University of Science and Technology of China, Hefei, China

[3] - Lucas Pugliese Barros, Flavio Medeiros, Eduardo Moraes, Anderson Feitosa Junior. Analyzing the Performance of Apps Developed by using Cross-Platform and Native Technologies. Informatics Coordination Federal Institute of Alagoas.

[4] – Vani Rajan. CNN BASED CLASSIFIER FOR IDENTIFICATION OF CANINE BREEDS. Seventh International Conference on Advances in Information Technology & Networking (ICATN'20) At: Coimbatore.

[5] אתר Merlin Bird ID

[6] אתר BirdNET

[7] אתר הצפרות הישראלי

[8] אתר לקסיקון מפה – המדריך השלם לציפורי אירופה וישראל