**מסמך פרויקט** [**xAI**](https://github.com/yotam180/xAI) **- תשע"ח**

**צוות:** יותם סלמון ושי קמחי.

**כיתה:** קריות 14.

**מדריך:** אלירן.

**מנטור:** לא תודה.

# פרק 1 – יזום

## תיאור כללי

פרויקט חקר המדגים את השימוש ברשתות נוירונים בתחום למידת המכונה (Machine Learning), ומביא לידי ביטוי את תחום הלמידה (Deep Learning) במימוש של זיהוי תמונות. הפרויקט יספק API למפתחים ולמשתמשי קצה כדי לזהות תמונות שונות בקטגוריות שונות, להרכיב ולאמן את רשתות הנוירונים שלהם בצורת REST API וללמד את המחשב לזהות תמונה בצורה נוחה דרך ממשק אינטרנט פשוט.

## מטרת הפרויקט

הפרויקט נועד להמחיש את היכולות של למידה עמוקה ושל רשתות נוירונים, ולתת הזדמנות לחוות את התחום למפתחים החדשים בתחום, למפתחים מנוסים שרוצים להיכנס לתחום הבינה המלאכותית, ואף למשתמשי קצה הרוצים ללמוד אותו.

פרויקטים דומים שקיימים בשוק:

* Google Cloud Vision API
* Clarifai
* Amazon Rekognition

## סוגי משתמשים – קהל היעד של היישום

בין האנשים היכולים למצוא שימוש בפרויקט ימצאו אנשי תוכנה צעירים שירצו ללמוד על הנושא של בינה מלאכותית ורשתות נוירונים, מתכנתים מנוסים שירצו להיכנס לתחום בצורה פשוטה, או אף משתמשי קצה שירצו להפעיל וללמוד מה קיים בתחום הבינה המלאכותית בצורה חברותית ולא מרתיעה. יש צורך ממשתמשי המוצר לידע בסיסי בתכנות מחשבים, אחרת המוצר עלול להיתפש רק כ"מגניב" ו"נחמד", אך המסר העיקרי לא יוכל לעבור בצורה משמעותית למשתמשיו הלא-מתכנתים.

## תהליכים עיקריים

* הוספת רשת חדשה – למשל, רשת חתול, רשת כלב, רשת מכונית (כל רשת תזהה את המושא שלה – כלב, חתול או מכונית, בדוגמאות שלעיל)
* הוספת תמונות לרשת – הוספת תמונות ספציפיות לרשת כדי שתוכל להתאמן עליהן וללמוד אותן.
* הוספת תמונות שליליות לרשת – אימון הרשת לאי-לקיחת התמונות כתמונות הקשורות לרשת.
* העלאת תמונה לזיהוי – ה-API יחזיר את תוצאות הרשתות הגבוהות ביותר שזיהו את התמונה כחיובית.
* העלאת "פידבק" על רשתות קיימות, ובכך אפשרות לביקורתיות על תוצאות הרשתות.

## טכנולוגיות עיקריות ושיקולים עיקריים

הטכנולוגיות העיקריות שסביר להניח שנשתמש בהן במהלך הפרויקט:

* Google TensorFlow™
* tfLearn (a TensorFlow library)
* NVIDIA Cura GPU SDK (for hardware acceleration)

## ידע, סיכונים ושאלות

בעת כתיבת המסמך, הידע היישומי שלנו בספריות רשתות נוירונים ובלמידת מכונה הוא מינימלי, על אף שהידע התיאורטי (המתמטיקה מאחורי קסמן של רשתות הנוירונים, עקרונות הפעולה שלהן וכו') קיים. נצטרך ללמוד את הצד הפרקטי של הנושא.

## פתרונות קיימים

[Google Cloud Vision API](https://cloud.google.com/vision/)

# פרק 2 – אפיון

במסמך זה יצוינו התהליכים העיקריים של xAI VISION.

## פיצ'רים ומסכים באתר ובגרסת ההדגמה של XAI

### הוספת רשת/קטגוריה

המשתמשים של XAI יוכלו להוסיף רשתות חדשות לזיהוי נושאי תמונות חדשים לשרתי XAI. בגרסת ההדגמה הויזואלית של XAI ובקונסולת המפתחים שבאתר, יוכלו המשתמשים להוסיף קטגוריה בצורה הבאה: המשתמשים ימלאו את מזהה הרשת שלהם ואת הנושא הטקסטואלי (למשל – שם רשת “yotam\_dog\_classifier\_1”, הנושא הטקסטואלי יהיה “dog”. המשתמשים יכניסו את הגדרתם לרשת (למשל – “A four legged mammal related to the wolves”). בשלב הבא, יכניסו המשתמשים מילות מפתח, ו-XAI יוציא אלפי תמונות מהאינטרנט בעזרת מילות המפתח האלה. למשל – Dog, Poodle, Dalmatic dog, cute dog, dogs, Labrador, bulldog, white dogs, black dog, וכו'. בנוסף, יכניסו המשתמשים מילים שליליות – שלא קשורות לקטגוריה שלהם. למשל –Cat, car, snake, house, forest, table, sign, person, alligator, water, rain, deer, laptop, וכו'. XAI יאסוף תמונות רבות של קטגוריות אלה גם כן, ויאמן את הרשת גם עליהן.

### הוספת תמונות לרשת

תהא אפשרות להוספת תמונות לרשת, הן דרך הממשק הויזואלי, דרך קונסולת המפתחים באתר XAI והן דרך ה-API הציבורי שהפלטפורמה תספק. כדי להעלות תמונות חדשות, כל שעל המשתמש יהיה לעשות הוא להעלות תמונה או קישור לתמונה באינטרנט, לבחור קטגוריה (רשת) ולסמן את חיוביות או שליליות התמונה (האם התמונה שייכת לקטגוריה או לא).

### אימון רשת

ניתן יהיה לאמן רשתות חדשות על קטגוריות שהועלו. הממשק יהיה מאוד פשוט, ויהיה ניתן להוסיף גרסאות של רשתות על כל תמונה.

### זיהוי תמונה

דרך גרסת ההדגמה באתר, או דרך ה-API עצמו, ניתן יהיה להעלות תמונה אל שרתי XAI ולקבל חזרה את התשובה מהרשתות הבולטות שסיווגו את התמונה בצורה הכי משמעותית בשבילן. הממשק הויזואלי באתר יספק אפשרות להעלאת התמונה והצגה ויזואלית (או בפורמט JSON) של תוצאות ההערכה. ב-API, התוצאה תתקבל כמערך אובייקטים.

### התחברות/הרשמה לאתר

באתר תהיה אפשרות להירשם עם תיבות טקסט המספקות את שם המשתמש, הסיסמה, אימות הסיסמה, דוא"ל, טלפון, מדינה (תיבת DropDown) וטקסט חופשי על עצמך. דרישות הקלט:

* Username must be at least 8 characters
* Username can be at max 30 characters
* Password must be at least 8 characters
* Password can be at max 50 characters
* Password must contain a number
* Password must contain an uppercase letter
* Password must contain a lowercase letter
* Password confirmation must be the same as the password
* The email must be a valid email address
* The Country combobox must hold an existing country

### פאנל אישי

מסך בו ניתן יהיה לראות רשימות קטגוריות ורשתות באתר של XAI שיצר המשתמש. המסך יהיה זמין רק למשתמשים מחוברים ולא תהיה בו אפשרות לעריכה. רק קישורים לדפי הקטגוריות ודפי הרשתות.

## ויזואליזצית ממשק

### קונסולת המפתחים של XAI

תפריט עליון

הוספת קטגוריה

הוספת תמונה לקטגוריה

הוספת רשת (מזהה)

זיהוי תמונה

## דרישות שאינן פונקציונליות

### בשימוש הממשק הגרפי

1. סביבת DOTNET להרצת הממשק הגרפי (הכתוב ב-#C)
2. חיבור אינטרנט

### בקונסולת המפתחים באתר האינטרנט

1. דפדפן תומך 5HTML (Chrome, Safari, Opera, Firefox, Edge, IE9+...)
2. חיבור אינטרנט
3. חיבור SSL בשביל אבטחת המידע

### בשביל השימוש ב-API

אחת מהדרישות הבאות:

1. פייתון 2.7 מותקן
2. פייתון 3.5 מותקן
3. ספריית DOTNET מותקנת (בשביל גרסת ה-#C של ה-API).

# פרק 3 - ארכיטקטורה

## דיאגרמת חלקים לפרויקט

## תיאור החלקים בפרויקט

### שרת ה-API

1. מודול תקשורת – שיהיה אחראי על התקשורת בין הלקוחות אליו לבין החלקים הפנימיים של ה-API. יטפל בהצפנות, ב-buffers ובכל הקשור לכניסת ויציאת מידע מהשרת של XAI.
2. מודול התחברות – יהיה אחראי על אימות המשתמשים, מפתחות ה-API, הכניסות והבקשות הנכנסות אל השרת.
3. מנהל מסד הנתונים – יהיה אחראי על עדכון ושליפה של נתונים מתוך מסד הנתונים של המשתמשים.
4. מסד הנתונים – קובץ קבצי JSON או שירכיבו את מסד הנתונים המקומי על מחשב השרת של XAI.
5. מודול עיבוד הבקשות הנכנסות – מודול שיעבד כל בקשה שנכנסת ויפעיל פקודות במנהל רשתות הנוירונים ומאגרי הנתונים בהתאם. יהיה אחראי על החזרת תשובה ללקוחות על בקשות שנעשו, ועל עדכון/שימוש ברשתות.
6. מנהל הרשתות ומאגרי הנתונים – יהיה אחראי על שליפה, אחזור ועדכון של נתונים, תמונות, רשתות נוירונים ומידע metadata ממאגרי המידע של XAI.
   1. ספריית nLib – הספרייה שפיתחנו ליצירת רשתות נוירונים מלאות תתפעל בשלב הראשון את מנהל הרשתות. לאחר מכן, נוסיף לה תמיכה בעזרת TensorFlow ו-NVIDIA GPU Drivers בכדי להאיץ את מהירויות החישובים ולהוסיף קונבולוציה של תמונות.
7. מאגרי המידע של XAI – מערכת הקבצים שתאחסן את כל המידע על הרשתות, התמונות והמידע של פלטפורמת XAI.

### שרת האינטרנט

1. מודול תקשורת – יהיה מובנה בתוך מערכת השרת ויהיה אחראי על התקשורת עם הלקוחות והצפנתה.
2. מודול עיצוב – יספק תבנית דפים אחידה וקובצי CSS לעיצוב דפי האינטרנט שבאתר.
3. מודול הלוגיקה הראשי – יהיה אחראי על עיבוד הבקשות, תקשורת עם שרת ה-API הפנימי בעבור התחברות, וידאג לתצוגה דינאמית של דפי האינטרנט שבאתר.

### לקוח ה-API

1. ממשק – הממשק שיחשף למפתחים כדי לגשת לפונקציונליות של ה-API.
2. ה-internals – אחראים על התקשורת עם שרת ה-API.

## עיצוב הנתונים וישויות מידע

סוגי המידע שישמרו במאגרים של שרת XAI:

* משתמש – מזהה משתמש, שם תצוגה, האש סיסמה, אימייל, טלפון, מדינה, דרך תשלום.
* מאגר מידע – מזהה מאגר, כותרת, משתמש שעל שמו רשום המאגר, תאריך יצירה, תאריך עדכון אחרון.
* מזהה (רשת) – מזהה רשת, שם, בעלים (משתמש), ביסוס (מאגר), תאריך יצירה, רמת דיוק.
* עדכון (למאגר) – מזהה עדכון, תוכן עדכון (הוספה/הסרה/שינוי), מזהה מאגר שעודכן, תאריך עדכון, האש קומיט.

## טכנולוגיות עיקריות

1. Python 3.5, בעזרתו יופעלו השרתים של XAI.
2. PHP להרצת האתר עצמו בצד השרת.
3. 5HTML, 3CSS, JS להרצת צד הלקוח של האתר.
4. #C, Python 3.5 או Python 2.7 להרצת לקוח ה-API.
5. ספריית NumPy להאצת החישובים המתמטיים של ספריית nLib, אותה ניצור.
6. ספריית TensorFlow בעבור קונבולוציה של תמונות והאצת החישובים של רשתות הנוירונים מעבר ליכולותיה של הספרייה שביכולתינו לספק.
7. ספריית tfLearn בעבור פישוט השימוש ב-TensorFlow.
8. ספריות של NVIDIA לעבודה על הכרטיס הגרפי בעבור האצת החישובים עוד יותר (Hardware Acceleration).

חשבנו להשתמש בצד השרת ב-++C כדי להאיץ את החישובים, אך הספריות של TensorFlow בנויות גם כך על C/++C כך שאין שום יתרון בכתיבת השרת והתקשורת ב-C גם כן. פייתון היא פשוטה ויותר אוניברסאלית ולכן בחרנו להשתמש בה בסופו של דבר. לגבי השרת, PHP היא הפלטפורמה המובילה והפופולארית ביותר בעולם, ואין עוררין על כך שלמטרותינו (אתר פשוט לויזואליזציית החיבור עם ה-API) היא תספיק בהחלט. נשתמש ב-Google Cloud כדי להריץ את האתר של XAI.

TensorFlow היא ספריית ברירת המחדל לשימוש ברשתות נוירונים עם פייתון, ואין ספק שיש לה עתיד מזהיר מכיוון שהיא מפותחת על ידי ענקית האינטרנט Google. יכולנו להשתמש בספריות אחרות כמו scikit learn או Theano, אך נראה שהשימוש ב-TensorFlow יחד עם המעטפת tfLearn הוא יותר פשוט ויעיל מהספריות האחרות.

## התאמה לאפיון

|  |  |
| --- | --- |
| הוספת מאגר (דרך אתר) | לקוח – ממשק אתר -> מערכת לוגיקה של אתר -> תקשורת שרת XAI -> בדיקת חיבור משתמש -> גישה למנהל מסד המשתמשים -> גישה למסד המשתמשים -> גישה למנהל הרשתות -> גישה למאגר הרשתות -> יצירת מאגר |
| הוספת מאגר (דרך API) | לקוח – ממשק API -> תקשורת שרת XAI -> בדיקת חיבור משתמש -> גישה למנהל מסד המשתמשים -> גישה למסד המשתמשים -> גישה למנהל הרשתות -> גישה למאגר הרשתות -> יצירת מאגר |
| הוספת תמונה לרשת (דרך אתר) | לקוח – ממשק אתר -> מערכת לוגיקה של אתר -> תקשורת שרת XAI -> בדיקת חיבור משתמש -> גישה למנהל מסד המשתמשים -> גישה למסד המשתמשים -> גישה למנהל הרשתות -> גישה למאגר הרשתות -> הוספת תמונה |
| הוספת תמונה לרשת (דרך API) | לקוח – ממשק API -> תקשורת שרת XAI -> בדיקת חיבור משתמש -> גישה למנהל מסד המשתמשים -> גישה למסד המשתמשים -> גישה למנהל הרשתות -> גישה למאגר הרשתות -> הוספת תמונה |
| אימון רשת (דרך אתר) | לקוח – ממשק אתר -> מערכת לוגיקה של אתר -> תקשורת שרת XAI -> בדיקת חיבור משתמש -> גישה למנהל מסד המשתמשים -> גישה למסד המשתמשים -> גישה למנהל הרשתות -> גישה למאגר הרשתות -> אימון |
| אימון רשת (דרך API) | לקוח – ממשק API -> תקשורת שרת XAI -> בדיקת חיבור משתמש -> גישה למנהל מסד המשתמשים -> גישה למסד המשתמשים -> גישה למנהל הרשתות -> גישה למאגר הרשתות -> אימון |
| זיהוי תמונה (אתר) | לקוח – ממשק אתר -> מערכת לוגיקה של אתר -> תקשורת שרת XAI -> גישה למנהל הרשתות -> גישה למאגר הרשתות -> קלסיפיקציה |
| זיהוי תמונה (דרך API) | לקוח – ממשק API -> תקשורת שרת XAI -> בדיקת חיבור משתמש -> גישה למנהל מסד המשתמשים -> גישה למסד המשתמשים -> גישה למנהל הרשתות -> גישה למאגר הרשתות -> קלסיפיקציה |
| הרשמה לאתר | לקוח – ממשק אתר -> מערכת לוגיקה של אתר -> תקשורת שרת XAI ->גישה למנהל מסד המשתמשים -> בדיקת נתונים והוספת משתמש |
| התחברות לאתר | לקוח – ממשק API -> תקשורת שרת XAI -> בדיקת חיבור משתמש -> גישה למנהל מסד המשתמשים -> גישה למסד המשתמשים |
| פאנל אישי | לקוח – ממשק API -> תקשורת שרת XAI -> בדיקת חיבור משתמש -> גישה למנהל מסד המשתמשים -> גישה למסד המשתמשים -> שליפת נתונים -> הצגת נתונים בעזרת מודול העיצוב |

# פרק 4 – חלוקת העבודה

## חלוקת חלקי העבודה

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **סדר** | **משימה** | **תלות** | **נימוק והערות** | **רכיב רלוונטי** |
| 1 | הכנת הרשת הבסיסית | אין | בניית הספרייה לחישובי הרשתות | Neural nets manager |
| 1 | הכנת mockup הממשק | אין | בניית הממשק המדגים |  |
| 2 | בניית מודול זיהוי התמונה בעזרת TensorFlow | הכנת הרשת הבסיסית | להאצת החישובים וכו' | Neural nets manager |
| 3 | מערכת ההתחברות באתר | אין | לבדיקת חיבור משתמשים וטוקנים. | User database manager |
| 3 | בניית רכיב התקשורת של השרת | אין | להתחברות של לקוחות אל שרת XAI. | Communication & SSL encryption module |
| 3 | בניית עיצוב האתר | אין | ממשק האתר. | Design module |
| 4 | בניית לוגיקת האתר PHP | בניית עיצוב האתר, בניית רכיב התקשורת של השרת | החיבור בין האתר לשרת של XAI. | Logic module PHP |
| 5 | חיבור חלקי השרת והפעלת ה-API הסופי. | בניית רכיב התקשורת של השרת, בניית מודול זיהוי התמונה | שרת עובד סופי אחרי החלק הזה |  |
| 6 | כתיבת ממשק ה-API. | חיבור חלקי השרת הסופי | פרויקט גמור. |  |

## חלוקה לאיטרציות

1. הרשת הבסיסית, מוקאפ הממשק, מודל זיהוי התמונה
2. מערכת ההתחברות בשרת, רכיב התקשורת, חיבור חלקי השרת.
3. עיצוב האתר, לוגיקת האתר ב-PHP.
4. ממשק ה-API.