

תיק פרויקט הגנת סייבר



מגיש: יובל מנדל (216496349( תיכון הרצוג כפר סבא יב'3

שם פרויקט: DigitNet

שם מנחה: אופיר שביט

שם חלופה: הגנת סייבר ומערכות הפעלה

תאריך הגשה:

תוכן עניינים

[מבוא 3](#_Toc197270421)

[ייזום 3](#_Toc197270422)

[תיאור כללי של המערכת 3](#_Toc197270423)

[הגדרת הלקוח 3](#_Toc197270424)

[מטרות ויעדים 4](#_Toc197270425)

[בעיות תועלות וחסכונות 4](#_Toc197270426)

[סקירת פתרונות קיימים 5](#_Toc197270427)

[טכנולוגיה 6](#_Toc197270428)

[תיחום 6](#_Toc197270429)

[פירוט תיאור המערכת 8](#_Toc197270430)

[תיאור מפורט והיכולותך שיש לכל רכיב 8](#_Toc197270431)

[פירוט הבדיקות (קופסא שחורה) 9](#_Toc197270432)

[לוח זמנים 11](#_Toc197270433)

[ניהול הסיכונים ודרכי התמודדות 12](#_Toc197270434)

[תיאור תחום הידע 13](#_Toc197270435)

[פירוט מעמיק של היכולות 13](#_Toc197270436)

[יכולות צד שרת 13](#_Toc197270437)

# מבוא

## ייזום

### תיאור כללי של המערכת

בפרויקט זה הינו מערכת שיכולה לזהות ספרות (0–9) מתוך תמונות, תוך שימוש בטכניקות של למידת מכונה, ובעיקר רשת נוירונים קונבולוציונית (CNN).

המערכת תכלול ממשק גרפי פשוט ונוח, שבו משתמשים יוכלו להעלות תמונות שכוללות ספרות, ולקבל את תוצאת הזיהוי בצורה של טקסט. בנוסף, תהיה אפשרות לצפות בתמונות שהועלו על ידי משתמשים אחרים, לראות איזו ספרה זוהתה בכל תמונה וגם להוריד את התמונות ששמורות בשרת.

התמונות והמידע הנלווה (כמו תוצאת הזיהוי) יישמרו במסד נתונים, כך שיהיה אפשר לשמור היסטוריה של הפעולות שהתבצעו במערכת. המערכת תשלב בין צד לקוח (GUI) שבו המשתמש פועל, לבין צד שרת שמטפל באימון המודל, בביצוע הזיהוי בפועל, ובניהול מסד הנתונים.

בחרתי בפרויקט הזה כי תחום הבינה המלאכותית מעניין אותי מאוד, במיוחד נושא של רשתות נוירונים ולמידה עמוקה. אני רוצה להבין איך רשת נוירונים מצליחה ללמוד לזהות תבניות מתוך מידע חזותי, ואיך אפשר ליישם את זה בפועל על בעיה אמיתית כמו זיהוי ספרות מתמונה. בנוסף, אני רואה בפרויקט הזה הזדמנות לשלב בין תיאוריה ומימוש מעשי, וללמוד על כל התהליך – החל מאיסוף הנתונים והכנתם, דרך אימון המודל, ועד לשילובו במערכת שלמה שפועלת מול משתמשים.

האתגרים שאני צופה לי בפרויקט הם אימון מדויק של מודל הבינה המלאכותית כך שהזיהוי יהיה אמין והבנה ומימוש של של רשתות נוירונים קונבולוציונית.

### הגדרת הלקוח

המערכת מיועדת לכל אדם שמעוניין לזהות ספרות מתוך תמונות בצורה מהירה, פשוטה ונגישה. היא מתאימה לשימושים חינוכיים, ניסיוניים או יישומיים — למשל לצורך זיהוי של ספרות שנכתבו ביד. קהל היעד כולל תלמידים, מורים, חוקרים ומפתחים המעוניינים לבדוק מודלים של למידת מכונה או להשתמש בתוצרי הזיהוי לצורכי ניסוי ולמידה.

המערכת מאפשרת למשתמשים להעלות תמונות, לקבל את תוצאת הזיהוי באופן מיידי, ולצפות בהיסטוריית הזיהויים שבוצעו בעבר. הממשק תוכנן להיות נוח לשימוש גם עבור מי שאין לו רקע טכני, והמערכת שמה דגש על חוויית משתמש פשוטה וברורה.

יתרון מרכזי של המערכת הוא באפשרות לאגור את התמונות שזוהו יחד עם תוצאות הזיהוי. מאגר זה משמש לא רק לתיעוד, אלא גם ככלי עזר ללמידת מכונה — ניתן לעשות בו שימוש חוזר לצורך אימון ושיפור של מודלים אחרים, מה שהופך את המערכת לכלי עבור תהליכים מתקדמים של ניתוח ולמידה.

### מטרות ויעדים

המטרות שלי הן לפתח מערכת שמסוגלת לזהות ספרות מתוך תמונות באופן אוטומטי בעזרת למידת מכונה. לממש ולהבין רשת נוירונים קונבולוציונית. לצבור ידע וניסיון מעשי בתחום של בינה מלאכותית וזיהוי תמונה.

### בעיות תועלות וחסכונות

הפרויקט עוסק בזיהוי אוטומטי של ספרות מתוך תמונות בעזרת בינה מלאכותית. מטרת המערכת היא לבצע את הזיהוי בצורה מהירה, מדויקת ויעילה, גם בסביבות עתירות נתונים ובתנאי קלט לא אחידים — כגון תמונות באיכות משתנה, עם רעש חזותי או כתב יד לא אחיד.

בנוסף לפונקציית הזיהוי הבסיסית, המערכת כוללת רכיב חשוב של אגירת מידע: כל תמונה שעוברת זיהוי נשמרת באופן שיטתי במאגר פנימי יחד עם תוצאת הסיווג ופרטים נוספים. המאגר הזה אינו רק אמצעי שמירה, אלא משמש תשתית ללמידה מתקדמת — ניתן לאמן על בסיסו מודלים חדשים או לשפר את המודל הקיים באמצעות הנתונים שהמערכת עצמה צברה.

באופן זה, המערכת לא רק מזהה ספרות מתמונה, אלא גם בונה בסיס נתונים שיכול לעזור באימון של מודלים אחרים בעתיד.

### סקירת פתרונות קיימים

הפרויקט שלי מתמקד בלפתור 2 בעיות, זיהוי ספרה ובניית מאגר תמונות יש להן שיוך למספר המתאים.

יש הרבה מוצרים שמסוגלים לזהות ספרות ואפילו מספרים מילים ומשפטים.

לדוגמה:  
<https://www.newocr.com/>

מערכת ocr שמסוגלת לזהות טקסט בתמונה (כולל ספרות)

האתר הזה נותן יותר יכולת מהפרויקט שלי (מסוגל טקסט בכללי).

|  |  |  |
| --- | --- | --- |
| תמונה | תוצאה של הפתרון הקיים | תוצאה שלי |
|  | 5 | 3 |
|  | SL | 7 |
|  | O) | 0 |

האתר אינטרנט נתן תוצאות לא נכונות.

הבדל נוסף בין הפתרון הזה לבין הפתרון שלי הוא שהפתרון הזה הוא עמוד web ובעוד שלי הוא שרת ולקוח עד פרוטוקול יחודי.

הבעיה השנייה שאוצתה אני פותר היא מאגר מידע, המאגר מידע של ספרות הכי פופולרי הואMNIST שבו אני משתמש כדי לאמן את המודל שלי. בעוד שמאגר MNIST כולל תמונות סינתטיות בפורמט אחיד — ספרות ממורכזות, ללא רעש, בעובי וקונטרסט קבועים — המאגר שאני בונה מורכב מתמונות קלט אמיתיות כפי שהועלו על ידי המשתמשים. לכן, הוא מייצג בצורה נאמנה יותר אתגרים מהעולם האמיתי ויכול לשמש ככלי עזר לmnist באימון של מודלים אחרים. (יתרון של mnist הוא שהוא מקוטלג על ידי בני אדם ולכן הסיכוי שיהיה טעות בזיהוי הוא אפסי לעומת המאגר שלי)

### טכנולוגיה

הטכנולוגיה שעליה מבוסס הפרויקט אינה חדשה, היא עושה שימוש ברשתות קונבולוציה- סוג של אלגוריתם בלמידת מכונה שמיועד במיוחד להבנה של תמונות. הרשת "לומדת" לזהות דפוסים וצורות חוזרות בתמונה, כמו קווים, עיקולים או מבנה של ספרות, בדיוק כפי שעין אנושית שמה לב לפרטים. בזכות היכולת הזו, רשתות קונבולוציה מתאימות במיוחד למשימות של זיהוי חזותי.

מכיוון שאני מממש את רשת ה־CNN מהיסוד וללא שימוש בספריות מתקדמות או מודלים מוכנים מראש, תהליך האימון פועל גבי מעבד (CPU) רגיל ולא על כרטיס גרפי (GPU) מה שמגביל את היכולת לבצע אימונים כבדים או להשתמש במודלים גדולים. בנוסף, מאחר שאין ברשותי חומרה חזקה במיוחד, נדרש איזון בין איכות המודל לזמן הריצה והיעילות. שני הגורמים האלו מגבילים את היכולת שלי לאמן מודלים גדולים וחזקים ופוגעים באיכות הסופית של המוצר. כדי להתגבר על המכשולים האלו ניסיתי לממש את הרשת בצורה יעילה.

### תיחום

הפרויקט עוסק במגוון של תחומים טכנולוגיים:

* למידת מכונה: הפרויקט מתמקד בזיהוי ספרות מתוך תמונות באמצעות רשתות נוירונים קונבולוציוניות תוך מימוש עצמי של תהליך האימון והחישוב, ללא שימוש במודלים מוכנים מראש.
* רשתות תקשורת: המערכת כוללת תקשורת בין לקוח לשרת באמצעות פרוטוקול TCP (בשביל אמינות בהעברת המידע) ומממשת פרוטוקול ייחודי להעברת בקשות וקבצים בצורה אמינה ויעילה. התקשורת תומכת בהעברת מידע בינארי (תמונות) ובניהול תהליך שליחה וקבלה בשכבת האפליקציה.
* הצפנה ואבטחת מידע: כל התקשורת בין הלקוח לשרת מוצפנת באמצעות AES (הצפנה סימטרית), כאשר מפתח ההצפנה מועבר בצורה מאובטחת באמצעות אלגוריתם RSA (הצפנה אסימטרית). שילוב זה מאפשר הגנה מלאה על המידע לאורך כל שלבי ההעברה.
* מערכות הפעלה: הפרויקט כולל שימוש ב- Threads לניהול תהליכים מקבילים בשרת ובלקוח, וכן עבודה מול מערכת הקבצים (קריאה, כתיבה ושמירה של תמונות ונתונים), כחלק מהאינטגרציה עם מערכת ההפעלה.

הפרויקט אינו עוסק ביצירת מערכת הרשאות או ממשק משתמשים עם שם משתמש וסיסמה. אין בו מנגנון התחברות, שמירת סיסמאות או ניהול משתמשים, והוא לא כולל אימות זהות של הלקוח.

## פירוט תיאור המערכת

### תיאור מפורט והיכולותך שיש לכל רכיב

המערכת נועדה לזהות ספרות (0–9) מתוך תמונות המועלות על ידי המשתמשים, ולהציג את תוצאת הזיהוי בצורה נגישה, מהירה וברורה. היא מבוססת על רשת נוירונים קונבולוציונית (CNN), שזו שיטה מתקדמת בלמידת מכונה המתמחה בניתוח מידע חזותי. רשת זו מסוגלת לזהות תבניות חזותיות מורכבות מתוך תמונה, גם כאשר הקלט אינו אחיד — כלומר, גם כאשר הספרה אינה ממורכזת, הקווים דקים או עבים מהרגיל, קיימים רעשים בתמונה, או שהתמונה צולמה או נסרקה באיכות משתנה.

אחד היעדים המרכזיים של המערכת, מעבר לזיהוי המדויק עצמו, הוא בניית מאגר תמונות ותיוגים שנוצרו מתוך קלטים אמיתיים של משתמשים. מאגר זה מהווה תשתית חשובה לאימון עתידי של מודלים נוספים, כך שהמערכת לא רק מזהה אלא גם **לומדת ומשתפרת** לאורך זמן. בכך, נוצרת פלטפורמה שמבוססת על נתוני אמת ולא רק על מאגר סטטי כמו MNIST, ומאפשרת גמישות והתאמה לשימושים רחבים יותר.

המערכת מחולקת לשני חלקים עיקריים:

**1. צד לקוח:**

חלק זה של המערכת הוא הממשק הגרפי שמולו פועל המשתמש. הוא נועד להיות פשוט, ברור ונגיש גם למי שאין לו רקע טכנולוגי קודם. הממשק מאפשר למשתמש להעלות קבצי תמונה מהמחשב האישי או מכל מקור אחר. לאחר ההעלאה, התמונה נשלחת לשרת, ותוצאת הזיהוי (הספרה שזוהתה) מוצגת מיד בצורה ברורה.

בנוסף לכך, הממשק כולל אפשרות לעיין בתמונות קודמות שעברו זיהוי — כולל תמונות שהועלו על ידי משתמשים אחרים. כל תמונה מוצגת יחד עם הספרה שזוהתה בה, וניתן גם להוריד את התמונות למחשב המקומי. תכונה זו מאפשרת למשתמשים לצפות בנתונים שכבר עברו דרך המערכת, לבחון את איכות הזיהוי ולהשתמש בתמונות לצרכים לימודיים או ניסיוניים.

יכולות: בקשה מהשרת לזהות תמונה, העלאת תמונה, הצגת תוצאה של השרת, הצגת/הורדת תמונות ותוצאות שנאגרו בשרת.

**2. צד שרת:**

חלק זה אחראי על עיבוד הנתונים. השרת מקבל את התמונות שנשלחות מהלקוח, מפענח את המידע ומכין אותו לקראת ניתוח. לאחר מכן, התמונה מועברת למודל ה- CNN שמבצע את תהליך החיזוי כלומר, קובע איזו ספרה מופיעה בתמונה שהתקבלה.

התוצאה מוחזרת ללקוח לצורך הצגה בממשק, אך גם נשמרת בצד השרת לצורך תיעוד וניתוח עתידי. כל תמונה מזוהה, יחד עם תוצאת הזיהוי שלה ופרטי עזר נוספים, נשמרים במסד נתונים פנימי. בכך המערכת בונה לעצמה באופן שוטף מאגר של דוגמאות מתויגות – בסיס נתונים חשוב שיכול לשמש לאימון נוסף, סטטיסטיקות, או מחקר עתידי בתחום הזיהוי החזותי.

יכולות: זיהוי ספרה בתמונה, שמירת תוצאות ותמונות, תקשורת ומתן שירות ללקוח

### פירוט הבדיקות (קופסא שחורה)

1. בדיקת העלאת תמונה תקינה

* מטרה: לבדוק האם המערכת מזהה תמונה שמכילה ספרה בודדת באופן תקין.
* אופן ביצוע: המשתמש יעלה תמונה איכותית, ברורה, של ספרה אחת (למשל, ספרה 3 כתובה בכתב יד ברור).
* תוצאה צפויה: המערכת תזהה את הספרה בצורה נכונה, והתוצאה תוצג בממשק ותישמר במסד הנתונים.

1. בדיקת העלאת תמונה באיכות נמוכה / עם רעש

* מטרה: לבדוק את עמידות המערכת לתמונות פחות איכותיות – לדוגמה, עם רעש, טשטוש קל או כתב יד לא ברור.
* אופן ביצוע: העלאת תמונה שצולמה מטלפון נייד בתאורה חלשה או עם רעש מלאכותי שנוסף לקובץ.
* תוצאה צפויה: המערכת תצליח לזהות את הספרה למרות הירידה באיכות.

1. בדיקת העלאת קובץ שאינו תמונה

* מטרה: לבדוק שהמערכת יודעת להתמודד עם קלט שגוי.
* אופן ביצוע: ניסיון להעלות קובץ מסוג PDF, קובץ טקסט או קובץ אקראי שאינו תמונה.
* תוצאה צפויה: המערכת תסרב לקבל את הקובץ ותציג הודעת שגיאה מתאימה (למשל: "סוג קובץ לא נתמך").

1. בדיקת תצוגת ההיסטוריה

* מטרה: לוודא שניתן לצפות בתמונות שזוהו בעבר יחד עם תוצאת הזיהוי שלהן.
* אופן ביצוע: לאחר מספר העלאות, המשתמש ילחץ על כפתור צפה בתוצאות קודמות.
* תוצאה צפויה: תוצג רשימה של תמונות קודמות עם תוצאת הזיהוי שלהן, כולל אפשרות להוריד כל אחת מהן.

1. בדיקת תגובתיות הממשק

* מטרה: לבדוק שהממשק הגרפי (GUI) מגיב בצורה חלקה לכל פעולה – העלאה, הצגה, מעבר בין כפתורים.
* אופן ביצוע: ביצוע רצף של פעולות משתמש – העלאה, מעבר להיסטוריה, חזרה למסך ראשי וכו'.
* תוצאה צפויה: כל פעולה תתבצע באופן מיידי ללא תקיעות או קריסות של הממשק.

1. בדיקת תקשורת עם השרת

* מטרה: לבדוק האם התקשורת בין הלקוח לשרת תקינה, והאם מתבצע תהליך שליחה וקבלה תקני.
* אופן ביצוע: ניתוק יזום של החיבור לרשת, או הרצת הלקוח כשהשרת כבוי.
* תוצאה צפויה: המערכת תציג הודעה על כשל בחיבור ולא תקרוס.

1. בדיקת אחסון במסד הנתונים

* מטרה: לבדוק שהמידע – תמונה, תוצאה וזמן – אכן נשמרים במאגר הנתונים לאחר כל זיהוי.
* אופן ביצוע: ביצוע זיהוי ולאחר מכן בדיקה (לוגית או תצוגתית) של הופעת הנתון החדש ברשימת ההיסטוריה.
* תוצאה צפויה: הנתון החדש יופיע עם כל הפרטים במיקום הנכון.

### לוח זמנים

|  |  |  |
| --- | --- | --- |
| משימה | תכנון | בפועל |
| חקר והבנה של תהליך למידת מכונה. להבין את השלבים בלמידה ובעיבוד של רשתות נוירונים Fully connected | 23.11.24-1.1.25 | 12.12.24-20.1.25 |
| מימוש של רשת Fully connected | 2.1.25-15.2.25 | 21.1.25 – 24.1.25 |
| להכין את השרת/לקוח(ממשק גרפי) + חיבור למסד נתונים | 20.2.25 – 3.4.25 | 20.2.25 – 3.4.25 |
| הבנה של רשת קונבולוציונית | 3.4.25 – 15.4.25 | 3.4.25 – 20.4.25 |
| מימוש של רשת קונבולוציונית | 15.4.25 – 30.4.25 | 20.4.25 – 2.5.25 |
| הוספה של פיצ'רים נוספים/שיפור של המערכת | 1.5.25-10.5.25 | 2.5.25 – 3.5.25 |

### ניהול הסיכונים ודרכי התמודדות

|  |  |  |
| --- | --- | --- |
| בעיה | דרך מוצעת להתמודדות | מה שנעשה בפועל |
| עיבוד תמונות ברזולוציות שונות | Resize לתמונה טרם העיבוד | Resize לתמונה טרם העיבוד |
| תפיסה של מקום גדול באחסון של השרת בגלל השמירה של תוצאות עבר | לשמור רק מספר קבוע של תמונות אחרונות | לשמור רק מספר קבוע של תמונות אחרונות |
| אי זיהוי תווים כתוצאה מודל לא מאומן מספיק | הרחבה של סט הנתונים | שינוי המבנה של המודל, אימון ארוך יותר והוספת מוטציות לתמונות על מנת לשפר generalization |
| קבצים גדולים מאוד עולים לקחת הרבה מקום באכסון של השרת | לשמור את התמונות אחרי resize | לשמור את התמונות אחרי resize |
| |  | | --- | | משתמש מעלה אותה תמונה מספר פעמים |  |  | | --- | |  | | מניעת כפילויות במסד הנתונים  על ידי hash של התוכן | מניעת כפילויות במסד הנתונים  על ידי hash של התוכן |
| קריסת הלקוח כשהשרת קורס | לסגור את הלקוח במקרה שהשרת קורס | להפריד בין הgui לתקשורת ובמקרה של התנתקות מהשרת לנסות להתחבר בחזרה כל כמה שניות |

# תיאור תחום הידע

## פירוט מעמיק של היכולות

### יכולות צד שרת

* 1. קבלת תמונות מהלקוח זיהוי התמונה ושליחת תוצאות בצורה מאובטחת-

לאפשר ללקוח לשלוח תמונה לשרת ולקבל תוצאת זיהוי – בצורה אמינה ומוצפנת.

פעולות/יכולות שנדרשות:

* הלקוח יתחבר לשרת באמצעות פרוטוקול TCP.
* שרת יקבל את התמונה (בבייטים).
* השרת יזהה את הספרה שבתמונה.
* לאחר זיהוי – שליחת התוצאה (ספרה + רמת ביטחון) חזרה ללקוח.
* כל זה יתבצע תוך הצפנה מאובטחת של המידע.

אובייקטים/מודולים נחוצים:

* TCP Socket
* הצפנה סימטרית (AES)
* החלפת מפתחות (RSA)
* פרוטוקול תקשורת אפליקטיבי
* encryptor/decryptor
* מודל לזיהוי ספרות מאומן
  1. עיבוד תמונה לזיהוי ספרה באמצעות מודל CNN-

עיבוד של תמונה שהתקבלה מהלקוח וביצוע זיהוי של הספרה שבתוכה באמצעות רשת נוירונים קונבולוציונית מאומנת.

**פעולות/יכולות שנדרשות:**

* המרת התמונה לפורמט אחיד (שחור-לבן, גודל קבוע, נירמול).
* המרת התמונה למערך מספרי שניתן להזין למודל.
* הפעלת המודל המאומן לקבלת תוצאה – הספרה שזוהתה.
* שמירה על מהירות תגובה סבירה לביצוע בזמן אמת.

**אובייקטים/מודולים נחוצים:**

* מודל CNN יעיל מאומן לזיהוי ספרות (לממש maxpooling, dense layer, conv layer ועוד)
* פונקציות preprocessing שמממירות את התמונה לפורמט מתאים ועושות שינוויים כדי לשפר את אחוזי ההצלחה בזיהוי
  1. שמירת תוצאות במסד נתונים-

שמירה של כל תוצאה של זיהוי במסד נתונים, כולל מידע כמו הספרה שזוהתה, רמת הביטחון, והקובץ עצמו, על מנת לאפשר שליפה עתידית ובקרה.

**פעולות/יכולות שנדרשות:**

* יצירת רשומה חדשה במסד הנתונים עבור כל תמונה.
* שמירה של מזהה ייחודי (UUID), תוצאה מזוהה, אחוז ביטחון, ונתיב לקובץ
* שמירה של קובץ התמונה עצמו במערכת הקבצים.
* הגבלת כמות התמונות השמורות (למשל ל־100 אחרונות בלבד).
* מניעת כפילויות באמצעות hash של תוכן התמונה.
* מחיקת קבצים ישנים שאינם קיימים יותר במסד הנתונים
* שליפה של תמונות לפי דרישות – לדוגמה לפי ספרה מסוימת.

**אובייקטים/מודולים נחוצים:**

* מסד נתונים (SQLite)
* מחלקה לניהול מסד הנתינים
* פונקציות לניהול קבצים ( read, open, write, delete)
* מחולל hash (SHA-256) למניעת כפילויות.
* שאילתות SQL לשליפה, עדכון ומחיקה.
  1. אימון מודל CNN לזיהוי ספרות-

לבנות ולאמן רשת נוירונים קונבולוציונית (CNN) מאפס, כך שתלמד לזהות ספרות מתוך תמונות באופן מדויק. האימון מתבצע על גבי מאגר נתונים הכולל תמונות מתויגות.

**פעולות/יכולות שנדרשות:**

* בניית ארכיטקטורה של רשת
* בניית כל החלקים שמרכיבים את הרשת (dense conv2d flatten maxpool)
* הגדרת פונקציית עלות (loss) מתאימה.
* ביצוע forward pass לקבלת חיזוי.
* חישוב backward pass (גרדיאנטים לפי השרשרת) לעדכון המשקלים.
* שימוש באלגוריתם אופטימיזציה (לדוגמה SGD)
* חלוקה ל־epochs ומעקב אחר שיפור באחוזי הדיוק.
* אפשרות לבצע augmentations (עיוותים אקראיים) בתמונות לשיפור היכולת להכליל.
* שמירה של המודל המאומן לשימוש עתידי.

**אובייקטים/מודולים נחוצים:**

* מחלקת CNN הכוללת את כל השכבות והפונקציות הדרושות.
* מחלקת optimizer
* נתונים לאימון (mnist)
* פונקציות serialization כדי לשמור את האימון שנעשה
* פונקציות שעושות עיוותים למידע המתקבל

### יכולות צד לקוח

* 1. יצירת חיבור מאובטח לשרת-

התחברות לשרת בצורה מאובטחת לצורך שליחה וקבלה של מידע רגיש (תמונות ותוצאות זיהוי).

פעולות/יכולות שנדרשות:

* פתיחת חיבור TCP לשרת.
* קבלת מפתח ציבורי מהשרת לצורך הצפנה אסימטרית (RSA).
* יצירת מפתח AES אקראי.
* הצפנת המפתח הסימטרי באמצעות RSA ושליחתו לשרת.
* מעבר לתקשורת מוצפנת ב-AES לכל ההודעות הבאות.

אובייקטים/מודולים נחוצים:

* TCP Socket
* RSA Encryptor
* AES Encryptor/Decryptor
* ניהול session להצפנה
* ממשק תקשורת מול השרת
  1. העלאת תמונה לשרת לצורך זיהוי-

לאפשר למשתמש לבחור תמונה מהמחשב ולשלוח אותה לשרת בצורה פשוטה ומוצפנת.

פעולות/יכולות שנדרשות:

* הצגת כפתור לבחירת קובץ מהמחשב.
* קריאת קובץ התמונה וטעינתו לזיכרון.
* בנייה של ההודעה לפי הפרוטוקול
* שליחת ההודעה לשרת במבנה פרוטוקול מוגדר מראש.
* המתנה לקבלת תשובה מהשרת.

אובייקטים/מודולים נחוצים:

* File chooser
* מנגנון קריאת קבצים - read()
* AES Encryptor
  1. קבלת תוצאות מהשרת והצגתן למשתמש-

להציג למשתמש את הספרה שזוהתה בתמונה ואת אחוז הדיוק של המודל, באופן ברור ונוח.

פעולות/יכולות שנדרשות:

* קבלת תגובת שרת בפורמט מוגדר (מוצפן).
* פענוח
* פירוק תוכן ההודעה – ספרה + אחוז ביטחון.
* הצגת התוצאה על גבי הממשק.

אובייקטים/מודולים נחוצים:

* AES Decryptor
* Protocol parser
* רכיבי GUI להצגת טקסט
* לוגיקת ניתוח הודעות
  1. הצגת היסטוריית תמונות ותוצאות-

לאפשר למשתמש לצפות בתמונות קודמות שזוהו בעבר (את כולן וסינון לפי ספרות) כולל תוצאה מזוהה ואפשרות להורדה.

פעולות/יכולות שנדרשות:

* שליחת בקשה לשרת לקבלת היסטוריית תמונות.
* קבלת רשימת תמונות + תוצאות.
* הצגת התמונות בתצוגת גלריה.
* הצגת הספרה שזוהתה בכל תמונה.
* אפשרות להוריד את התמונות

אובייקטים/מודולים נחוצים:

* התמונות שהתקבלו מהבקשה
* מציד תמונות תמונות (image viewer)
  1. הורדת תמונות מהשרת-

לאפשר למשתמש להוריד למחשב האישי שלו את התמונות שנשלחו וזוהו בעבר.

פעולות/יכולות שנדרשות:

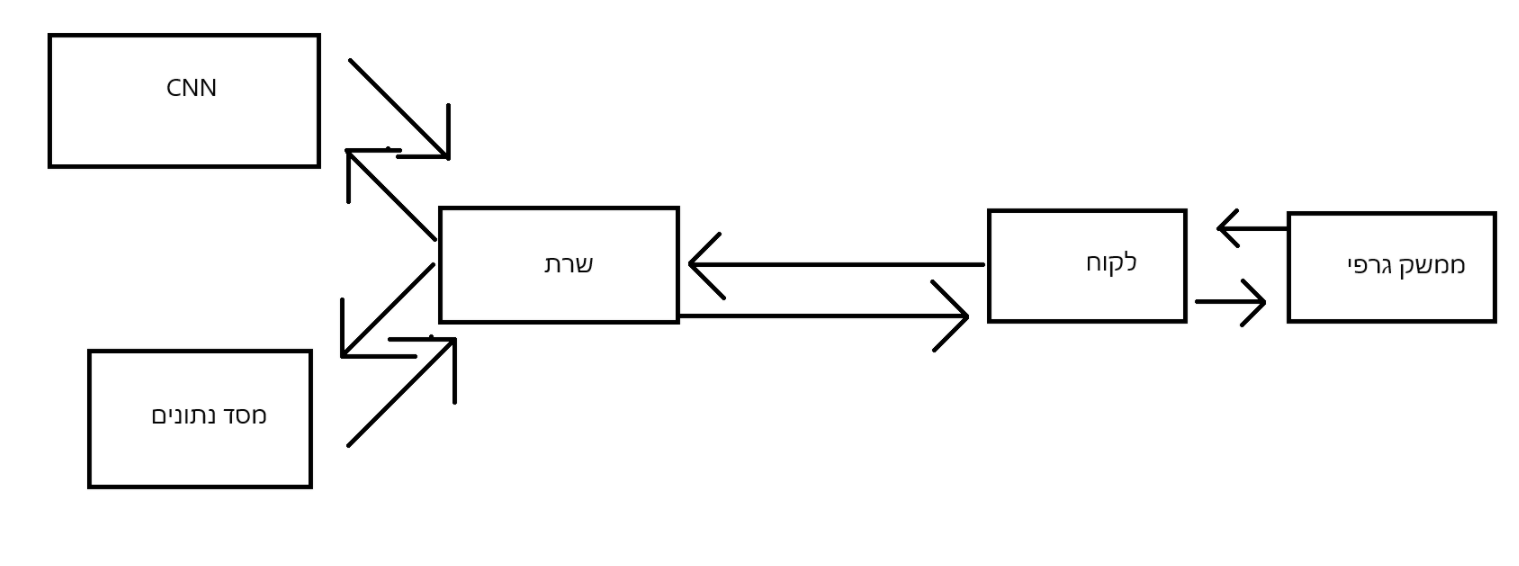
* הצגת כפתור "הורד" ליד כל תמונה בהיסטוריה.
* שליחת בקשה לשרת לקבל הקבצים וקבלתם.
* שמירה של הקובץ במחשב המקומי בפורמט מתאים.

אובייקטים/מודולים נחוצים:

* File dialog לבחירת תיקייה
* יוצר קבצי zip

# מבנה הפרויקט

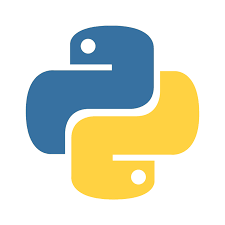
## תיאור הארכיטקטוקרה

לפרויקט 5 רכיבים עיקריים: ממשק גרפי, לקוח, שרת, רשת נוירונים קונבולוציונית ומסד נתונים

## תיאור הטכנולוגיה הרלוונטית

כל הקוד בפרויקט נכתב בשפת python (בפרט כדי לכתוב את ה- CNN השתמשתי בספריית Numpy).

בחרתי לכתוב את הפרויקט בשפת פייתון מכיוון שהיא שפה פשוטה, קריאה וגמישה, שמתאימה במיוחד לפרויקטים שמשלבים תקשורת. לפייתון יש קהילה רחבה וכלים מוכנים לעבודה עם תמונות ומערכים רב מימדיים בצורה יעילה (Numpy, PIL). היא מתאימה מאוד לפרויקטים לימודיים שמטרתם הבנה של הלוגיקה מאחורי המימוש – ולא רק שימוש בספריות מוכנות מראש.



השרת והלקוח רצים במערכת הפעלה windows 11.

התקשורת במערכת מנוהלת בפרוטוקול TCP בשביל אמינות של המידע. בנוסף התקשורת פועלת לפי פרוטוקול יעודי שהגדרתי למען הפרויקט.

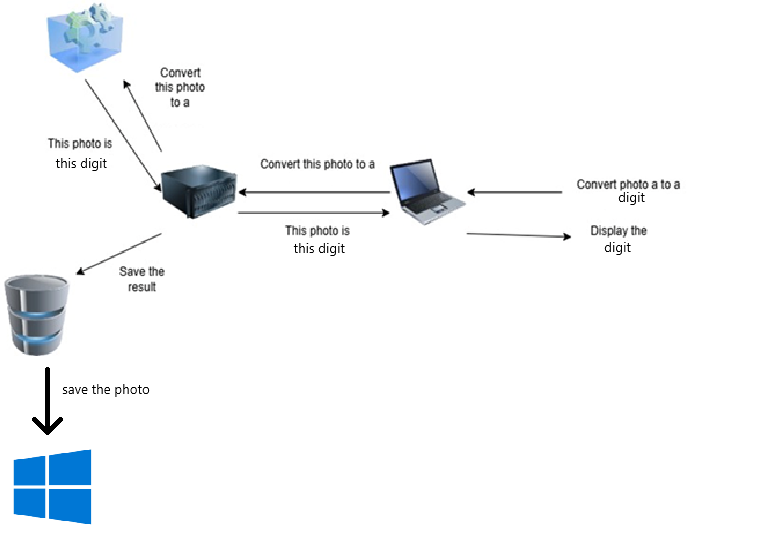
הפרויקט עוסק בשילוב של תחומים טכנולוגיים מגוונים: למידת מכונה וזיהוי תבניות חזותיות בעזרת רשתות נוירונים קונבולוציוניות (CNN), הצפנה ואבטחת מידע (RSA, AES), ופיתוח מערכות מבוססות לקוח-שרת עם תקשורת אמינה בפרוטוקול TCP.

בנוסף, הוא כולל ניהול נתונים באמצעות מסד נתונים מקומי, עבודה עם קבצים ותמיכה בממשק גרפי אינטראקטיבי. תחומי עניין אלו מבטאים חיבור בין מדעי המחשב העיוניים לבין יישומים מעשיים של תוכנה ובינה מלאכותית.

## תיאור זרימת המידע במערכת

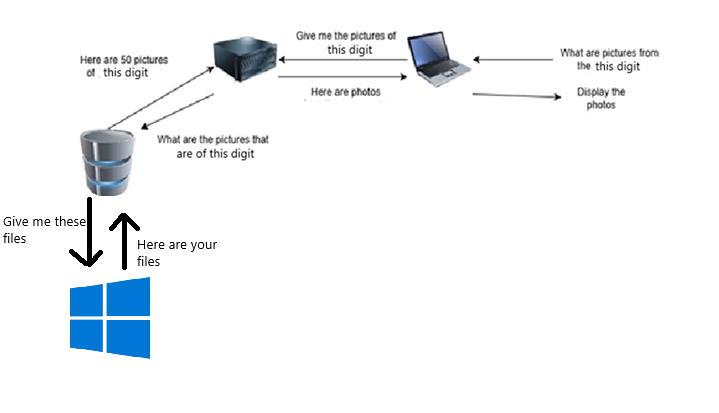
בקשה של הלקוח לפענח ספרה מסויימת:

המשתמש יזין בממשק הגרפי תמונה שהוא רוצה להמיר, הממשק הגרפי יודיע ללקוח לשלוח הודעה לשרת, הלקוח ישלח את התמונה לשרת לפיענוח, השרת ישתמש בCNN כדי לפענח את התמונה, ישמור במסד הנתונים את התוצאה (שישמור את התמונה מול מערכת ההפעלה) וישלח את התוצאה ללקוח שיציג את התוצאה בממשק הגרפי של המשתמש.



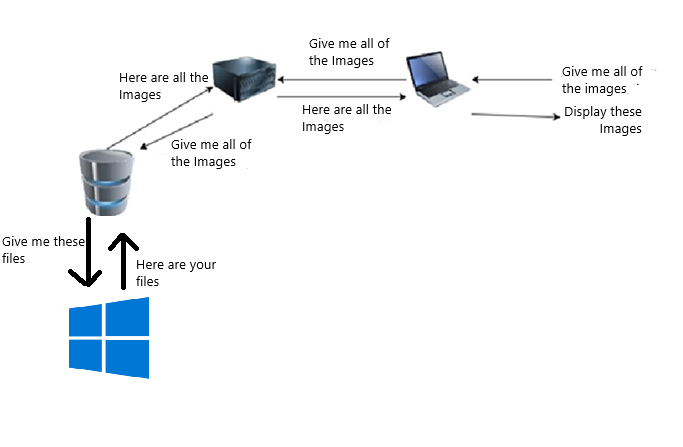
בקשה של הלקוח של תמונות של ספרה מסויימת:

המשתמש יבקש תוצאות קודמות של ספרה מסויימת דרך הממשק הגרפי, הממשק הגרפי יודיע ללקוח לשלוח הודעה לשרת, הלקוח יבקש מהשרת תשובות קודמות, השרת ישלוף ממסד הנתונים את התוצאות הקודמות, השרת ישלח את התוצאות ללקוח שיציג את התוצאות בממשק הגרפי.



בקשה של הלקוח של כל התמונות שיש לשרת:

המשתמש יבקש את כל התמונות שיש לשרת דרך הממשק הגרפי, הממשק הגרפי יודיע ללקוח לשלוח הודעה לשרת, הלקוח יבקש מהשרת תשובות קודמות, השרת ישלוף ממסד הנתונים את התוצאות הקודמות, השרת ישלח את התוצאות ללקוח שיציג את התוצאות בממשק הגרפי.



המשתמש רוצה להוריד את כל הקבים שנשלחו לו (יכול לקרות רק אחרי שקיבל תמונות (בשני הדרכים שצויינו לעיל)

