Azrieli
College of Engineering
Jerusalem

עזריאלי
מכללה אקדמית להנדסה
ירושלים

המחלקה להנדסת תוכנה

תיוג תמונות לפי הקטיגוריה המכלילה ביותר

חיבור זה מהווה חלק מהדרישות לקבלת

תואר ראשון בהנדסה

מאת

סיף אדין פרחאן

תשרי- חשון תשע"ט **יוני 2019**

עמוד השער הפנימי - דף דוגמה

המחלקה להנדסת תוכנה

תיוג תמונות לפי הקטיגוריה המכלילה ביותר

חיבור זה מהווה חלק מהדרישות לקבלת

תואר ראשון בהנדסה

מאת

סיף אדין פרחאן

(חתימה)

מנחה אקדמי: ד"ר אללוף מרים	**אישור:**	**תאריך:**
אחראי תעשייתי: ד"ר אללוף מרים	**אישור:**	**תאריך:**
רכז הפרויקטים: ד"ר שפנייר אסף	**אישור:**	**תאריך:**

תקציר

כחלק מלימודי הנדסת תוכנה לתואר ראשון בעזריאלי - המכללה האקדמית להנדסה ירושלים, ביצעתי פרויקט גמר מחקרי בהנחיית ד״ר מרים אללוף.

המטרה העיקרית של הפרויקט הייתה לקבל מטא-דאטה ממאגר התמונות המאוחסן במסד ה ElasticSearch ולנתח אותו ולחלק את התמונות לפי התוכן.

ישנו קושי רב בניתוח וחלוקה של מאגר עצום של תמונות לתמונות לפי תכנים דומים. לכן בפרוייקט זה במקום לנתח את התמונה , יש ניתוח של המילים המתארות את התמונה המאפשר הנגשה טובה יותר לתמונות במאגר.
אין דרך פשוטה לאתר תמונות בעלות תוכן דומה מלבד חיפוש ידני עד אשר כל התמונות המשותפות באותו תוכן נמצאות.

לשם כך בפרויקט זה בניתי מנוע המנתח בעזרת טכניקות NLP (עיבוד שפה טבעית) ולמידת מכונה אוספי מילים המתארות תמונה ומזהה את המילים החשובות והמגדירות את התמונה בצורה הטובה ביותר .
הכלי הותקן בשרת המכללה המאפשר גישה מרחוק ונבדק על מאגרים מסוגים שונים.

הצהרה:

העבודה נעשתה בהנחיית ד"ר מרים אללוף במחלקה להנדסת תוכנה, עזריאלי– המכללה האקדמית להנדסה ירושלים.
החיבור מציג את עבודתי האישית ומהווה חלק מהדרישות לקבלת תואר ראשון בהנדסה.

קישורים למערכות ניהול הפרויקט ובקרת תצורה

#	מערכת	מיקום
1	מאגר קוד	https://github.com/SeifAldenFarhan/Tagging-images-by-the-most-inclusive-category
2	יומן	https://github.com/SeifAldenFarhan/Tagging-images-by-the-most-inclusive-category/wiki/Project-Diary
3	סרטון גיסרת אלפא	https://www.dropbox.com/s/5u01smttvgfbi08/vedio.mp4?dl=0

תוכן העניינים

1. תיאור מסגרת הפרויקט

פרויקט זה הינו פרויקט מחקרי המאפשר לחלק מאגר תמונות לקבוצות של תמונות דומות על פי התגים הטקסטואליים המתארים את האובייקטים הנמצאים בתמונות אלו.
מטרת הפרויקט שלי היא להשתמש במילים ומטא-דאטה המתארים את כל התמונות כדי לסווג את אוסף התמונות לקבוצות של תמונות דומות על פי המילים והתוכן המשותף להן, שיעשה את החיפוש יותר קל על המשתמש. כדי לזהות דמיון בין אוספים של מילים המתארים תמונה השתמשתי בטכניקות של עיבוד שפה טבעית **Natural Language Processing (NLP).** NLP הוא סט של כלים המאפשרים להוציא מידע מובנה משפה טבעית על מנת שנוכל להבין אותה בתוכנה. מדובר באלגוריתמים של Machine Learning המבוססים על מודלים סטטיסטים המסוגלים למצוא תבניות בטקסט משפה טבעית.
בדרך כלל ניתן לזהות דמיון בין מסמכים במאגר בעזרת המילים הדומות שנמצאות במסמכים. לשם כך כל מילה במסמך מזוהה בעזרת מדד הנקרא **TF-IDF**. מדד זה הינו קיצור של **Term Frequency – Inverse Document Frequency**. בטכניקת אינדוקס מתקדמת זו מנוע החיפוש מצמיד ערך לכל אחת ממילות המפתח במסמך לפי כמות הפעמים שהיא מופיעה מסמך ובטקסטים אחרים, בעלי ערך, שנבדקו מראש. הערך הזה מאפשר למנוע החיפוש לצפות כמה פעמים מילה מסוימת תופיע בטקסט כתוב היטב, וכך להביל בין טקסטים בעלי ערך שבהם מילת מפתח מסוימת מופיעה מעט פעמים לטקסטים גרועים שבהם מילת המפתח מופיעה הרבה פעמים ללא סיבה, ובין טקסטים שמזכירים את הנושא אבל לא מתעמקים בו.
בפרויקט זה אנו מניחים שכל תמונה מלווה באוסף של תגים (terms) - המהווה מסמך בפני עצמו. אני מוצא לכל תמונה את וקטור מדדי ה TF-IDF הרלוונטיים לכל מילות התמונה ומחלק את מאגר התמונות למספר קבוצות של תמונות דומות בעזרת פונקציית ה Clustering K-Means. אני מציג לכל קבוצת תמונות דומות את התמונות בקבוצה ומאגר המילים שבה בצורת WordCloud.
Clustering: מתייחס למשימה של קיבוץ אובייקטים לקבוצות (אשכולות) כך שהאובייקטים הנמצאים באותה קבוצה דומים זה לזה יותר מאשר לאובייקטים השייכים לקבוצות אחרות.
פרוייקט זה הינו פרוייקט המשך לפרוייקט גמר של יונתן ידיד שנעשה שנה שעברה בספריה הלאומית, הפרוייקט של יונתן ידיד התייחס למאגר גדול של צילומים של הצלם בן הדני, תייג את התמונות בעזרת קריאות לשירות ה Google Vision ושמר את המטאדאטה ב- **ElasticSearch**.
בפרוייקט שלי כדי לבצע NLP על התמונות, אני משתמש בתשתית זו ובתיאורי התמונות השמורים ב - ElasticSearch.

2. תיאור הבעיה

כידוע כיום מערכות מחשבים יכולות לכלול כמויות גדולות של מידע. על מנת לנהל את כל המידע בצורה יעילה, ישנה חשיבות רבה לדרך שבה המידע שמור, ממוין ומשותף, וכמו כן לדרך שבה נערך חיפוש על המידע חשוב לדאוג איך נציג המידע.

תגיות הן מילות מפתח אשר משמשות לתאר חתיכת מידע, בין אם מדובר בדף אינטרנט, תמונה דיגיטלית או סוג אחר של מסמך דיגיטלי. שיטה זו מקלה על הדילמה כיצד לקטלג את הפריטים כאשר כל פריט יכול להיות מתויג בתגיות המתאימות לו. הכוח הגדול שבתגיות בא לידי ביטוי כאשר מתבצע חיפוש תמונות. כעת אין צורך לזכור תיקייה ספציפית שבה נמצאת התמונה, אלא לחשוב על תגיות מסוימות אשר עשויות לתאר את התמונה. בחיפוש אחר תגית מסוימת יופיעו כל התמונות שהיא משויכת להן.

יצירת מנוע חיפוש תמונות דומות על פי תוכנן לא רלוונטית במצב זה, מאחר ואין דרך לזהות או לאתר תמונות מסוימות במאגר. החיפוש היחיד שניתן אולי לממש במצב הנתון הוא על פי אוסף מאוד מצומצם של מילים המייצגות את תוכן תתי המאגרים. בפרויקט שלי אני מנצל את המידע על מידע (מטה-דאטה) לארגון וחיפוש יעילים של התמונות,

המידע הקיים הוא שמות התמונות ומטא-דאטא שנמצא ב Elasticsearch , והארגון הוא חלוקה לתתי מאגרים עם תוכן דומה (למשל תמונות של מפות או תמונות שחור לבן).כדי להסביר איך נזהה שתי תמונות דומות עפ״י אוסף המילים המתארות אותן אשתמש בדוגמאות הבאות :

- התמונה הבאה PIC1

מתוארת במילים

battle, history Cossacks, infantry, military organization, soldier, troop, rebellion

מילים אלו נשלפו ע״י חבילת ה Google Vision

(https://vision.googleapis.com/v1/images:annotate) ומתארות את תוכן התמונה.

כרגע אוסף המילים שמור ב ElasticSearch לפי ID.

- התמונה הבאה PIC2

מתוארת במילים

infantry, military organization, vehicle, soldier, troop crew

אנחנו יכולים לראות שהמילים soldier ו- troop מופיעות בתמונה הראשונה והשנייה. ומאפיינת אותן. ואילו המילה vehicle מופיעה רק בתמונה השנייה ואופינית רק לה. אם מספר המילים הנמצא בשתי התמונות הינו גדול נוכל להגיד שהתמונות דומות.

מדד ה IDF-TF אומר את הדבר הבא:

TF = number of times the term appears in the doc/total number of words in the doc

IDF = ln (number of docs/number docs the term appears in)

TF-IDF = TF * IDF

** Higher the TFIDF score, the rarer the term is and vice-versa.

למשל PIC2:

TF (vehicle, PIC2) = 1 / 7 = 0.142

IDF (vehicle, PIC2) = ln (2/1) = 0.693 # suppose I have two pictures

TF-IDF (vehicle, PIC2) = 0.142 * 0.693 = 0.0984

דרישות ואפיון הבעיה

להלן דרישות המחקר בפרוייקט זה:

- לחקור וללמוד את הטכניקות הבאות ולהבין איך הן ממומשות מעל ה ElasticSearch
- ניתוח: לנתח את המידע (המיטא-דאטה) של כל התמונות, ולהדגיש על המילים החשובים ביותר.
- פיצול: לחלק מאגר התמונות למספר של תתי קבוצות של תמונות דומות על פי התוצאות של הניתוח של המיטא-דאטא.
- הצגת: להציג קבוצת התמונות בחלון נפרד אחרי בחירת קבוצה מסויית.

הבעיה מבחינת הנדסת תוכנה

הפרויקט כולל הרבה מאוד אתגרים אלגוריתמיים וטכנולוגיים שלקח לי זמן רב להבין ולחקור אותם:

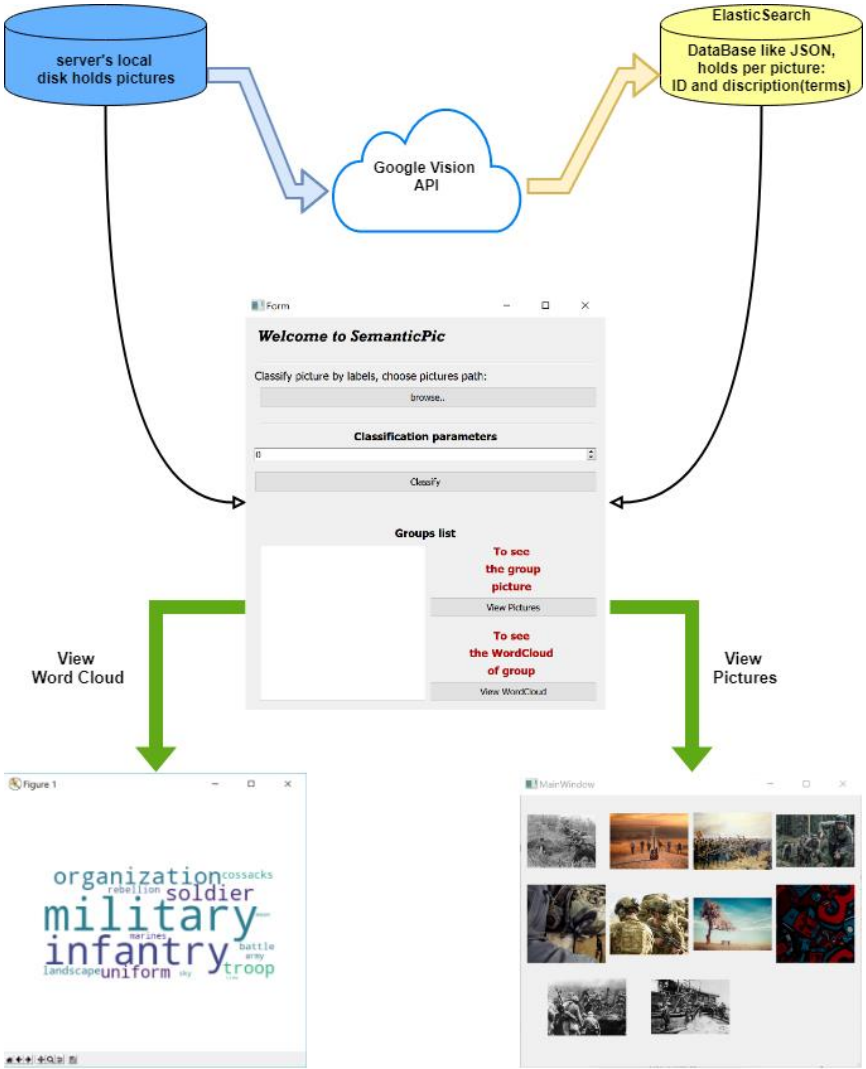
1. ElasticSearch & Kibana
2. שימוש בספריות פייתון כדי להשתמש במתאם לדמיון בין מסמכים והשוואת המסמך לאוסף המילים עבור התמונה – בנושאי NLP, IDF-TF ועוד.
3. לחקור וללמוד איך מבצעים clustering מעל מטא-דאטה של תמונות ושימוש ב K-means כדי לחלק ל-K קבוצות
4. הצגת המילים בעזרת WordCloud and JSON

כמו שציינתי לפני, אלה הם כלים חדשים עבורי – והייתי חייב ללמוד להפעילם כדי להצליח בפרויקט. יש המון מונחים שחייב לחקור וללמוד איך להשתמש בהכלים אלה לטובת הפרויקט.

3. תיאור הפתרון

ארכיטקטורת המערכת

התרשים הבא מתאר את ארכיטקטורת ופונקציות המערכת. זוהי מערכת ייחודית בעלת ממשק משתמש פשוט ונוח שלא צריך ידע טכני מיוחד, כך שכל בעל תפקיד מורשה יכול להשתמש בה.
המערכת זמינה בכל רגע נתון על השרת המערכת מאפשרת למשתמש לבחור את PATH ממנו יילקחו התמונות. לחיצה על כפתור הקבצים שרוצה לייצר מחלוקת התמונות ומגוון אפשריות מאיזה מאגר תמונות לפי סוג האחסון (אובייקטים, סמלילים -לוגואים-, טקסט ומיקומו בתמונה, פרצופים ומיקומם בתמונה, ישויות מוכרות או אתרים מוכרים).
המערכת אחראית גם על הצגת תתי הקבוצות בחלון חדש ומסודרים, והצגת המילים (התגיות) של כל קבוצה לפני להחליט אם כן להציג התמונות שלה או לא.

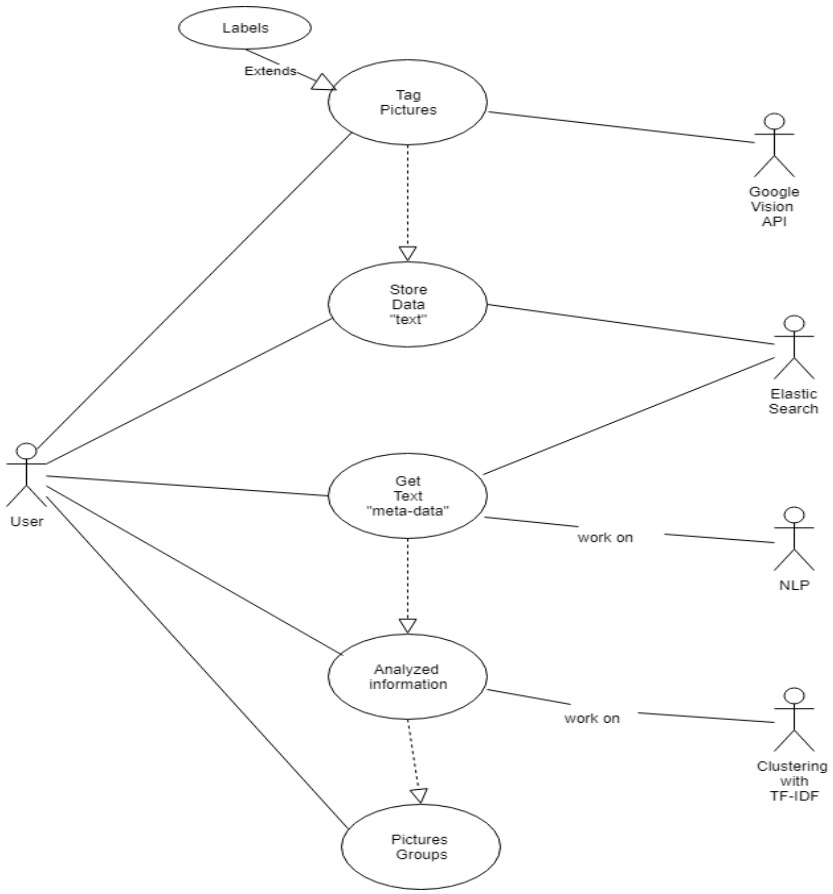


השיקולים שנלקחו בחשבון בבניית הארכיטקטורה הם מודולריות של הקוד והפרדה לתפקידים, לכל מודול צריך להיות תפקיד מוגדר ללא העמסה של קוד מיותר, בניסיון לבנות את המערכת בצורה גמישה שמאפשרת עדכון קוד קל במידת הצורך.

תהליכים ופעולות המערכת

המערכת כוללת תהליך אחד בכל זמן נתון, ועל גבי תהליך זה מתבצעות מגוון הפעולות. בכל בחירה של המשתמש לביצוע פעולה כלשהי, התהליך מבצע את הפעולה והמערכת ממתינה לסיום הפעולה, כאשר בסיום הפעולה המערכת מתפנה לביצוע פעולה נוספת ביחד עם הפעולה שקם לה, כאילו לבחור לראות יותר מקבוצה בחלונות שונות באותו רגע.

התרשים הבא מתאר את אינטראקציית המשתמש עם המערכת.



פונקציות המערכת מורכבת ממספר מודולים:

getDataEs.py – המודל האחראי על אוסף המטא-דאטה. הוא מתחבר ל- ElasticSearch ואוסף כל המטא-דאטה של התמונות ושומר אותם על הדיסק.

tfidf.py – המודל האחראי על הפעלת אלגוריתם TF-IDF על המטא-דאטה באופן הבא:

אופן ביצוע פיצול התמונות

קודם כל, מנתחים המידע ששמרנו בדיסק, מוחקים כל ה Stop Words ע״י ה- NLP.

אח״כ, מפעיל אלגוריתם ה- TF-IDF על המידע ומקבלים טבלה של כל המילים (תגיות) עם ה- TF-IDF score לכל תמונה.

A ו- B הם קבצים (Documents) או לפי הפרויקט הם תמונות.

Word	TF		IDF	TF*IDF	
	A	B		A	B
The	1/7	1/7	log(2/2) = 0	0	0
Car	1/7	0	log(2/1) = 0.3	0.043	0
Truck	0	1/7	log(2/1) = 0.3	0	0.043
Is	1/7	1/7	log(2/2) = 0	0	0
Driven	1/7	1/7	log(2/2) = 0	0	0
On	1/7	1/7	log(2/2) = 0	0	0
The	1/7	1/7	log(2/2) = 0	0	0
Road	1/7	0	log(2/1) = 0.3	0.043	0
Highway	0	1/7	log(2/1) = 0.3	0	0.043

בסוף, בוחרים מספר תתי קבוצות ועושים Clustering על התגיות והמשקל שלהם שקבלנו בטבלת ה- TF-IDF, ומשייך כל תמונה לקבוצת התגיות המתאימה לה ביותר.

Groups.py – המודל אחראי על חלוקת כל התמונות לתתי קבוצות. המודול משתמש ב K-means מעברים מספר תתי הקבוצות שרוצים, ומסתמך על תוצאות ה- TF-IDF בכדי לפצל המטא-דאטה ואח"כ התמונות עצמם לקבוצות דומות בתוכן.

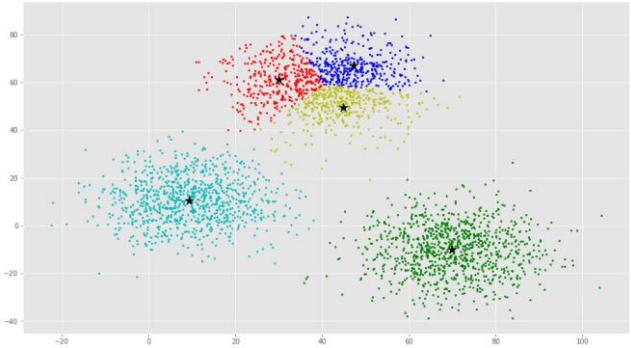
<u>Clustering</u>: מתייחס למשימה של קיבוץ אובייקטים לקבוצות (אשכולות) כך שהאובייקטים הנמצאים באותה קבוצה דומים זה לזה יותר מאשר לאובייקטים השייכים לקבוצות אחרות (K-means).

<u>אלגוריתם K-means:</u>

נגדיר קבוצה ראשונית של k מרכזים ואז האלגוריתם ינוע לסירוגין בין שני שלבים הבאים:

- **שלב ההקצאה:**
  נחשב לכל תצפית באשכול את המרכז הקרוב אליה לפי המרחק האוקלידי מהמרכזים שנבחרו, זה באופן אינטואיטיבי מרכז הכובד "הקרוב ביותר" (מבחינה מתמטית, זה אומר שמתקיימת חלוקת תצפיות לפי דיאגרמת וורונוי שנוצרת על פי המרכזים)

- **שלב העדכון:**
  נחשב את המרכזים מחדש על פי המרחק האוקלידי מהנקודות שהוקצו אליהם.



להבנה עמוקה יותר (https://www.youtube.com/watch?v=_aWzGGNrcic)

directoryPath.py – המודל האחראי על בחירת כתובת התיקיה של התמונות (Path), פותח חלון לבחור תקיה מסוימת של תמונות כי לא לעשות חלוקה לכל התמונות שנמצאות ב-ElasticSearch.

beta.py – הבקרה (Control) של שאר המודלים לאסוף ולהעביר המידע, ומחובר למודלים של הממשק הגרפי (View).

firstPageGui.py – אובייקט המייצג את העמוד העיקרי בממשק משתמש גרפי, שדרכו בוחרים התקיה של התמונות ומסבר הקבוצות, ואח"כ מתבצע אוספת המידע ואלגוריתם IDF-TF והחלוקה לתתי קבוצות. בסוף בוחרים קבוצה מסויימת להציג את התמונות שלה או להציג התגיות ששיכות לה בעזרת ה- WordCloud.

secondPageGui.py – אובייקט המייצג את התמונות של קבוצה מסויימת בחלון נפרד.

תרשים מס' 1 ו-2 בפרק 10 מתאר את הפונקציות והתרשים הפונקציונלי .

עלויות ומגבלות של שימוש בממשק

- אין עלות כספית לשימוש במערכת, אני משתמש בשרת שמותקן במכללת עזראילי לשמירת המידע.
- ממשק תומך במספר גדול של תמונות, אבל ככל שמספר המידע יגדל הממשק יהייה יותר איטית.
- הממשק לא מוגבל מבחינת מספר קבוצות ופתיחת חלונות לתתי קבוצות או חלונות של WordCloud.
-

אופן ביצוע הוספת המידע

המערכת יוצרת חיבור לשרת ומתקשרת ל- port של ה- ES שהוא 9200, ואח"כ יש צורך לשלוח למסד הנתונים חבילה שמכילה את האינדקס שבו המידע שמור (labels). החבילה החוזרת תכיל את המידע השמור בתוך אותו אינדקס, ושומרים את המידע בקובץ text ביחד עם שמות התמונות. פעולת השליפה במסד הנתונים מתבצעות על ידי REST API, את חבילת HTTP הנדרשת לשימוש השליפה במסד הנתונים היא HTTP GET.

אופן ביצוע הצגת התמונות וה- WordCloud

בוחרים קבוצה להציג התמונות שלה, מציגים רק התמונות ששיכות לתקיה שבחרנו, ומציג אותם בחלון חדש באותו גודל של כל התמונות שאני קובע בכדי שיהיו מסודרים ונוח לבדיקה.
לגבי ה-WordCloud גם פותחים חלון חדש ומציג התגיות בגודל שונה לפי ה-TF-IDF score והצבעים שונים.

תיאור הכלים ששימשו לפתרון

בפרויקט בשתמשתי במספר כלים לפתרון:

ElasticSearch: מנוע חיפוש המבוסס על Lucene (ספריית תוכנה לאיתור ואיסוף ידע).
הוא מספק, בין היתר, אינספור אופטימיזציות על מנת לספק ביצועים מהירים מאוד מאפשר לבצע אנאליזות על המסמכים, סכימה גמישה והתקנה מבוזרת. זהו מנוע חיפוש עוצמתי וגמיש שמסוגל להתמודד עם כמויות מידע עצומות.

Kibana: היא מערכת ויזואליזציה ואנליזה המנתחת נתונים ומספקת יכולות בינה עסקית, בוסס קוד פתוח, ונכתבה בשפתJavaScript., המערכת היא נדבך חשוב מסט הכלים של Elasticsearch, תומכת במגוון מערכות הפעלה: Windows, לינוקס, ומותאמת באופן מלא לעבודה עם Logstash ו-Elasticsearch.

TF-IDF: אחת הטכניקות המתקדמות ביותר היא חיפוש ממוצאות TF-IDF.
הביטוי, שהוא קיצור של (term frequency – inverse document frequency) מתאר טכניקת אינדוקס מתקדמת שבה מנוע החיפוש מצמיד ערך לכל אחת ממילות המפתח לפי כמות הפעמים שהיא מופיעה בממוצע בטקסטים אחרים, בעלי ערך, שנבדקו מראש. הערך הזה מאפשר למנוע החיפוש לצפות כמה פעמים מילה מסויימת תופיע בטקסט כתוב היטב, וכך להביל בין טקסטים בעלי ערך שבהם מילת מפתח מסויימת מופיעה מעט פעמים לטקסטים גרועים שבהם מילת המפתח מופיעה הרבה פעמים ללא סיבה, ובין טקסטים שמזכירים את הנושא אבל לא מתעמקים בו.

NLP: עיבוד שפה טבעית – Natural Language Processing, NLP – הוא סט של כלים המאפשרים להוציא מידע מובנה משפה טבעית על מנת שנוכל להבין אותה בתוכנה. מדובר באלגוריתמים של Machine Learning המבוססים על מודלים סטטיסטים המסוגלים למצוא תבניות בטקסט משפה טבעית.

מחשב בעל **מערכת הפעלה** **Windows 10** עם חיבור אינטרנט.

כתיבת הקוד נעשתה בשפת **פייתון (Python)** גירסה 3.7 .

4. תיאור המערכת שמומשה

תוכנה בעלת ממשק גרפי פשוט, נקי, וקל לתפעול. התוכנה כוללת שלוש חלונות שלכל אחד מהם יש ייעוד מסוים.

החלון הראשי:

בהחלון הזה המשתמש יכול לבחור התקיה המכילה התמונות שרצים לפצל אותם, ולבחור מספר הקבוצות שרוצים לקבל, ואח"כ המשתמש יכול לבחור אחת קבוצה להציג התמונות שלה או ה- Word Cloud שלה.

חלון התמונות:

החלון הזה מכיל את התמונות ששיכות לקבוצה מסוימת שהמשתמש בחר מהחלון הראשי, הצגת התמונות היא בשורות ובאותו גודל לכל התמונות.

חלון ה- Word Cloud:

החלון הזה מכיל את ה- Word Cloud ששיכות לקבוצה מסוימת שהמשתמש בחר מהחלון הראשי, הצגת ה- Word Cloud היא בגודל שונה לפי ה- TF-IDF score של כל מילה, ככל שהמילה חשובה יותר, כך גודלה גדול יותר, ובצבעים שונים.

5. תכנית בדיקות

כאמור המערכת מורכבת ממספר מודולים האחראים על תפקידים שונים. הבדיקות הראויות לביצוע בפרויקט זה הן:

- בדיקות יחידה - בדיקות עבור הפונקציות השונות במודולים, על מנת להבטיח שהפונקציות עושות את מה שהן אמורות לעשות, ולא עושות את מה שהן לא אמורות לעשות.
- בדיקות אינטגרציה - בדיקות הכוללות את תקינות ונכונות הפונקציות של המודולים השונים בשימוש במודולים אחרים.
- בדיקות ממשק לקוח (GUI) - בדיקות חווית הממשק הגרפי ותפקודו.

סוג הבדיקה	תיאור הבדיקה	תוצאות הבדיקה
יחיה	הפעלת השיטה לכתוב המיטא-דאטה לקובץ קיים.	השיטה דורסת המידע שיש בקובץ וכותבת עליו מחדש.
יחידה	הפעלת השיטה לכתוב המיטא-דאטה לקובץ לא קיים.	השיטה מייצר קובץ חדש וכותבת עליו.
יחידה	הפעלת השיטה לפצל התמונות.	השיטה עובד טוב ומפצלת התמונות לפי התוכן.
יחידה	הפעלת השיטה לבחור Directory path	השיטה מחזירה הכתובת של התקיה ולא חייב לבחור file מסוג מסוים.
מסד נתונים	הפעלת השיטה לשליפת נתונים.	הנתונים נשמרים בפורמט JSON המבוקש, הנתונים מוחזרים בפורמט JSON המבוקש, הנתונים נמחקים, ומתקבל ערך False בניסיון לשלוף שוב.
מסד נתונים	ניסיון שליפה\חיפוש\מחיקה מאינדקס לא קיים.	מתקבלת שגיאה שהאינדקס לא קיים והפעולות לא מתבצעות.
ממשק לקוח (GUI)	בדיקת לוגיקה והתנהגות תקינה של הרכיבים בעמוד הראשי.	לחיצה לקבל ה- directory path שומרת הכתובת במשתנה לשימוש אחר״כ, הכנסת מספר הקבוצות עובר לשיטת ה-Clustering.
ממשק לקוח (GUI)	בדיקת הצגה מדויקת והלוגיקה של מספר הקבוצות והזמינות לשינוי בכל רגע.	לחיצה על כפתור הפיצול מיצר קבוצות שכל אחת מחוברת לאוסף התמונות שלה, ואפשר לשינוי מספר הקבוצות ללא צורך להפעלת התוכנית מחדש.
ממשק לקוח (GUI)	בדיקת הצגת התמונות בהעמוד השני אחרי לחצה על View (pictures).	החלון מציג התמונות בסדר וגודל שווים לכל השורות.

ממשק לקוח (GUI)	בדיקת הצגת ה- Word Cloud.	ההצגה של התגיות היא בגודל שונה לפי החשיביות של כל מילה.

6. סקר שוק והשוואה

בפרק זה אדון במוצרים והשירותים הקיימים בשוק, וברמת הדמיון והרלוונטיות שלהם לפרויקט.

אפליקציית גוגל תמונות: האפליקציה משמשת כגלריית תמונות, והיא מספקת אפשרויות עריכה שונות של תמונות, אנימציות, אלבומים ועוד. בנוסף יש גיבוי של התמונות בענן וארגון אוטומטי של התמונות. דמיון לפרויקט: התמונות באפליקציה ניתנות למציאה על ידי חיפוש אנשים, מקומות או אובייקטים המופיעים בהן ללא צורך בתיוג פיזי. מידת התאמה לפרויקט: תיוג התמונות אוטומטי ואינו ניתן לשליטה מלאה ולהתאמה אישית, אין אפשרות לדעת או לבחור את התגיות של כל תמונה. גם הגיבוי בענן הוא אוטומטי ואינו ניתן להתאמה אישית. לכן אין אפשרות להשתמש באפליקציה זו לצורך הפרויקט, לא בבחירת תגיות ולא בבניית מסד נתונים.

חברות מיקור המונים (Crowdsourcing): חברות אשר משתמשות ב"כוח הקהל" לביצוע עבודות במגוון תחומים. מיקור המונים הוא הפניה של ביצוע משימה או משימות אשר לרוב היו מתבצעות בידי עובדי חברה או ארגון, לביצוע על ידי קהל גדול. חברת מיקור המונים יכולה להציע משימה לקהל הרחב ולהעניק תשלום למבצע המשימה הטוב ביותר, או לחלק את המשימה ואת התשלום בין מספר מבצעים שונים. ההתפתחות הטכנולוגית מאפשרת גישה במקביל למספר גדול של אנשים באמצעות רשת האינטרנט.

כך למשל החברה "Clickworker" מקבלת על עצמה פרויקטים של תיוג תמונות. החברה מקבלת מהלקוח מפרט דרישות הכולל בין השאר את כמות התמונות שהוא מעוניין לתייג, את שפת התיוג ואת כמות התיוגים לתמונה. החברה מחלקת את הפרויקט לעבודות קטנות ומפרסמת אותן לקהל הפרילנסרים המתאים שלה. מידת התאמה לפרויקט: ניתן לפנות לחברה לצורך ביצוע תיוג התמונות, אך מדובר בבינה אנושית ותיוג פיזי של התמונות על ידי אנשים שונים, כך שאין בהכרח תיאום בשמות התגיות. כמו כן, בהוספה של תמונות חדשות למאגר הקיים יהיה צורך בפניה נוספת לחברה. בנוסף אין פתרון לאחסון המידע.

תוכנות לארגון תמונות: קיימות כיום תוכנות כדוגמת ACDSee 20 או Zoner Photo Studio שמאפשרות ארגון יעיל יותר של התמונות בין השאר על ידי הוספת תגיות או מיקומים. מידת

התאמה לפרויקט: שוב מדובר בתיוג פיזי ולא בדרך תכנותית. התוכנות פועלות על קבצים המאוחסנים בזיכרון המקומי או בזיכרון נייד, כלומר יש צורך לייבא את התמונות כעבודה מקדימה.

לסיכום, המוצרים הקיימים כיום אכן מביאים לידי שימוש את הקונספט של תיוג אך אינם מהווים פתרון מספק לדרישה זו. נקודה חשובה נוספת היא נושא האחסון וחיפוש לפי תגיות שונות: מוצרים הקיימים אינם מהווים פתרון בנושא הביג דאטה. הם אינם בנויים לאחסון ולניתוח כמויות מידע גדולות, ולעמידה באתגרים של כתיבה ושליפה מהירות מאוד. לאחר סקירה מקיפה ניתן לומר שאין מוצר העונה על דרישות הפרויקט במלואן. הפרויקט ייחודי בדרישות שלו ודורש גמישות רבה, תאמה אישית ושליטה מלאה בהתנהלות, ואינם מהווים אפשרות לחפש ולחלק על תמונות מסוימות לפי פרמטר מסוים.

7. מסקנות מהפרויקט

שהתחלתי לעבוד על הפרויקט היה לי חוסר הבנה על הפרויקט וחוסר ידע כי אני הולך ללמוד דברים חדשים וכלים שלא השתמשתי לפני. למדתי הרבה מהפרויקט, בתחום הטכני והמקצועי וכן בתחום הניהולי והארגוני. התנסיתי בלימוד עצמאי וביישום של נושאים לא מוכרים כמו שפת תכנות חדשה, ממשק פיתוח ומסד נתונים לא מוכרים. למדתי איך להשתמש בכלים קיימים שרכשתי בלימודים ואיך לשלב עם כלים חדשים.

נוכחתי להבין באופן מעשי שתכנון מוקדם, ניהול זמן וניהול משימות הם תנאים הכרחיים להצלחה של פרויקט. למדתי איך להתאים עבודה ומימוש לדרישות ותיאור של לקוח (מנחה הפרויקט).

באשר לנושא הפרויקט עצמו, גיליתי שלמידע על מידע (מטה-דאטה) יש יתרונות גדולים בארגון של קבצים ובחיפוש אחר קבצים, ולהבין מה המשותף בין הקבצים השונים, במיוחד בתקופה המודרנית שכמויות מידע ושטחי אחסון הולכים וגדלים, שזה נותן כוח למנוע חיפוש מהיר לפי מילי מפתח (תגים) לחפש בתוך תתי קבוצות משאר לחפש בהקבוצה הגדולה. וגם לחלק מידע עצום לקבוצות קטנות לפי פרמיטרים שונים או לפי קטיגוריה מסויימת. לדעתי, נושא המטה-דאטה ימשיך להתפתח ויהיה חלק אינטגרלי במערכות ממוחשבות.

8. ספרות

- Python 3.7 וספריות חשובות

http://python.org/download/release/python-37

http://docs.python.org/3

http://docs.python-request.org/en/master

http://docs.python.org/3/library/json.html

http://docs.python.org/3/library/io.html

http://docs.python.org/3/library/sys.html

https://jakevdp.github.io/PythonDataScienceHandbook/05.11-k-means.html

https://amueller.github.io/word_cloud/

https://doc.qt.io/qtforpython/

- ElasticSearch

https://www.elastic.co

https://www.elastic.co/guide/en/elasticsearch/reference/current/docker.html

https://www.udemy.com/elasticsearch-6-and-elastic-stack-in-depth-and-hands-on/

- TF-IDF

https://towardsdatascience.com/tfidf-for-piece-of-text-in-python-43feccaa74f8

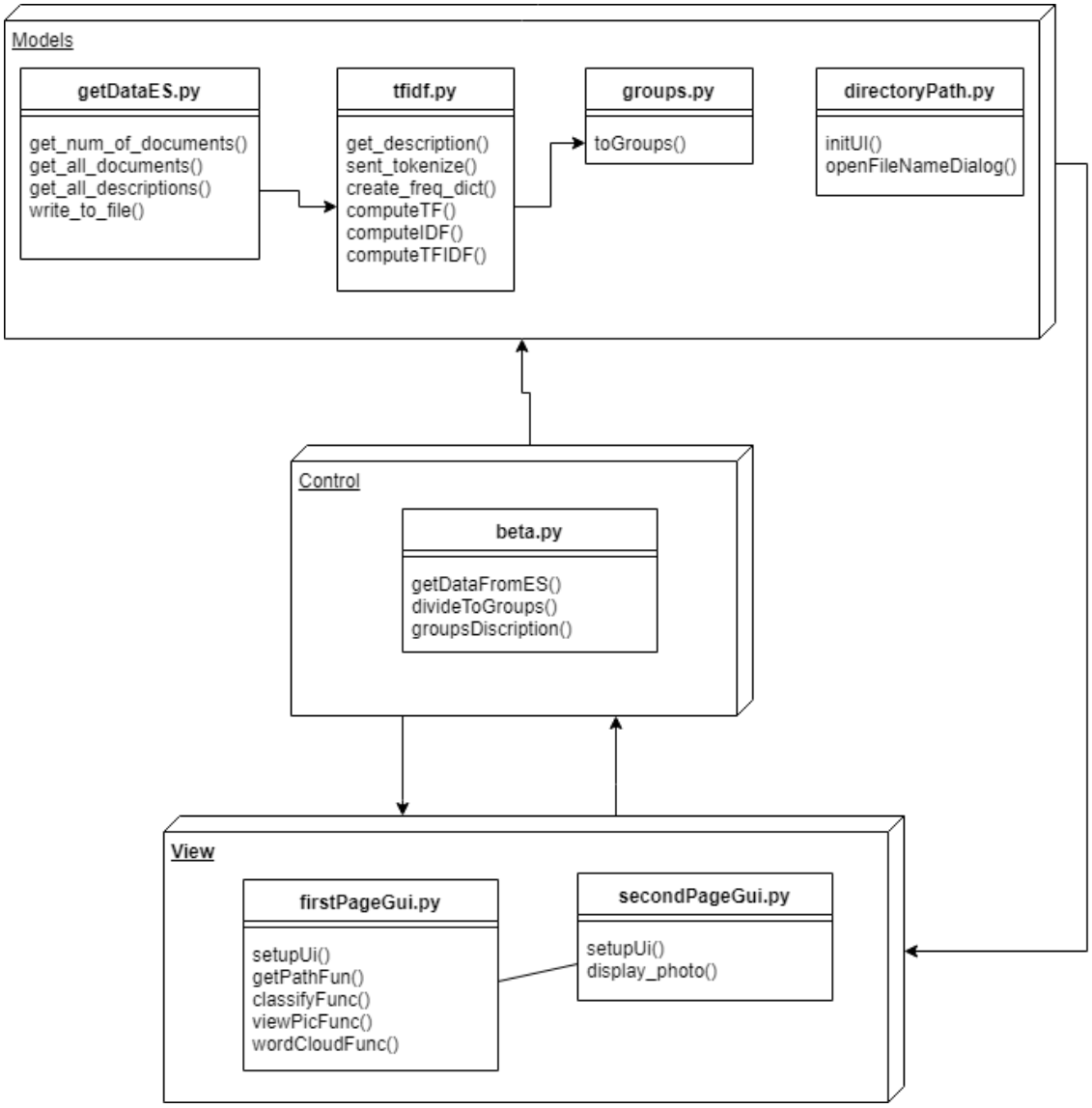
https://www.elephate.com/blog/what-is-tf-idf/

9. נספחים

<u>תרשים מס' 1 – ארכיטקטורת המערכת</u>

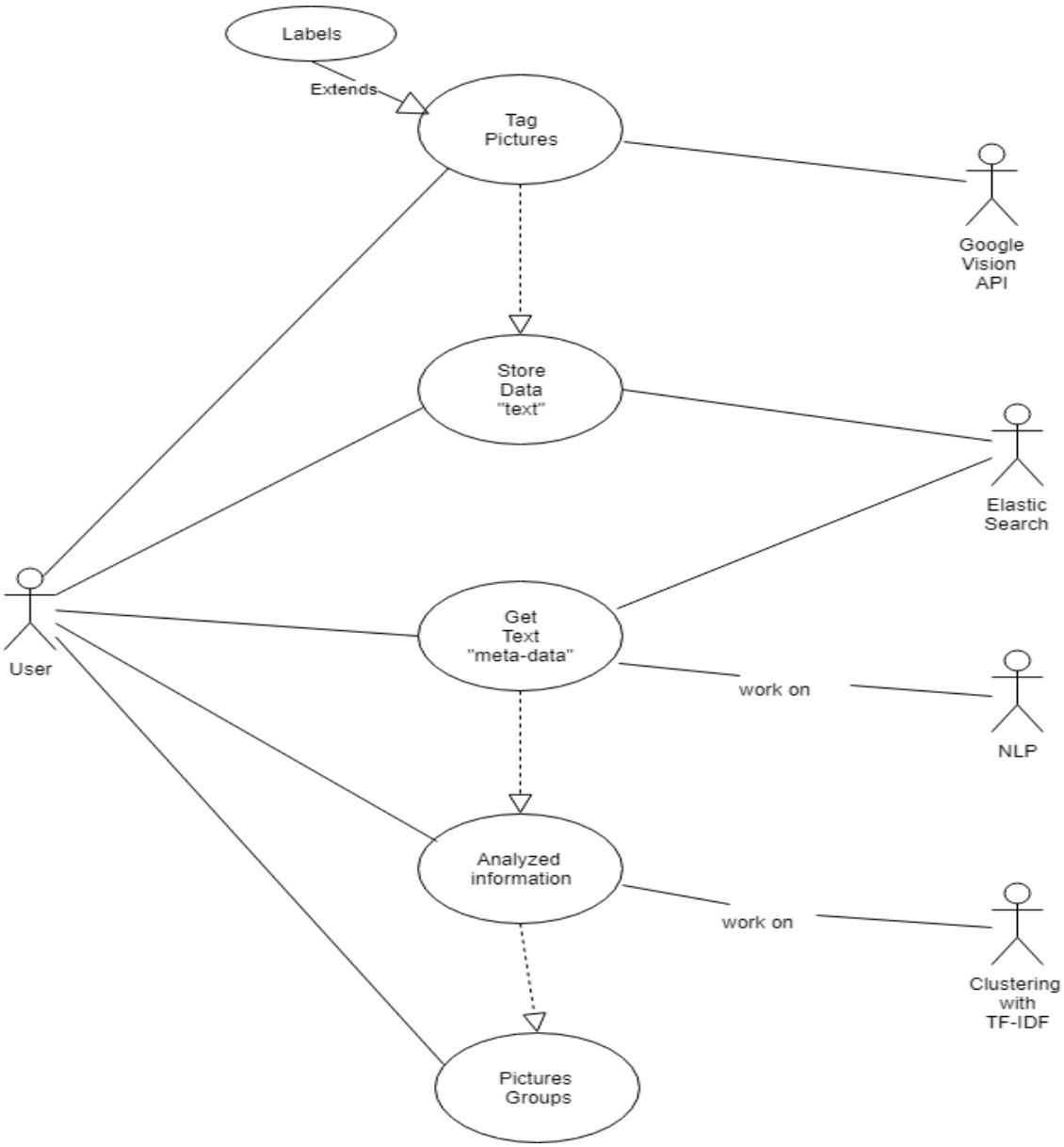
התרשים מתאר את מבנה הקבצים של המערכת ואת השימושים שלהם בין המערכת.

בתרשים מתוארות גם השיטות הממומשות בקבצים (רק שם השיטה ללא פרמטרים או ערכים חוזרים).



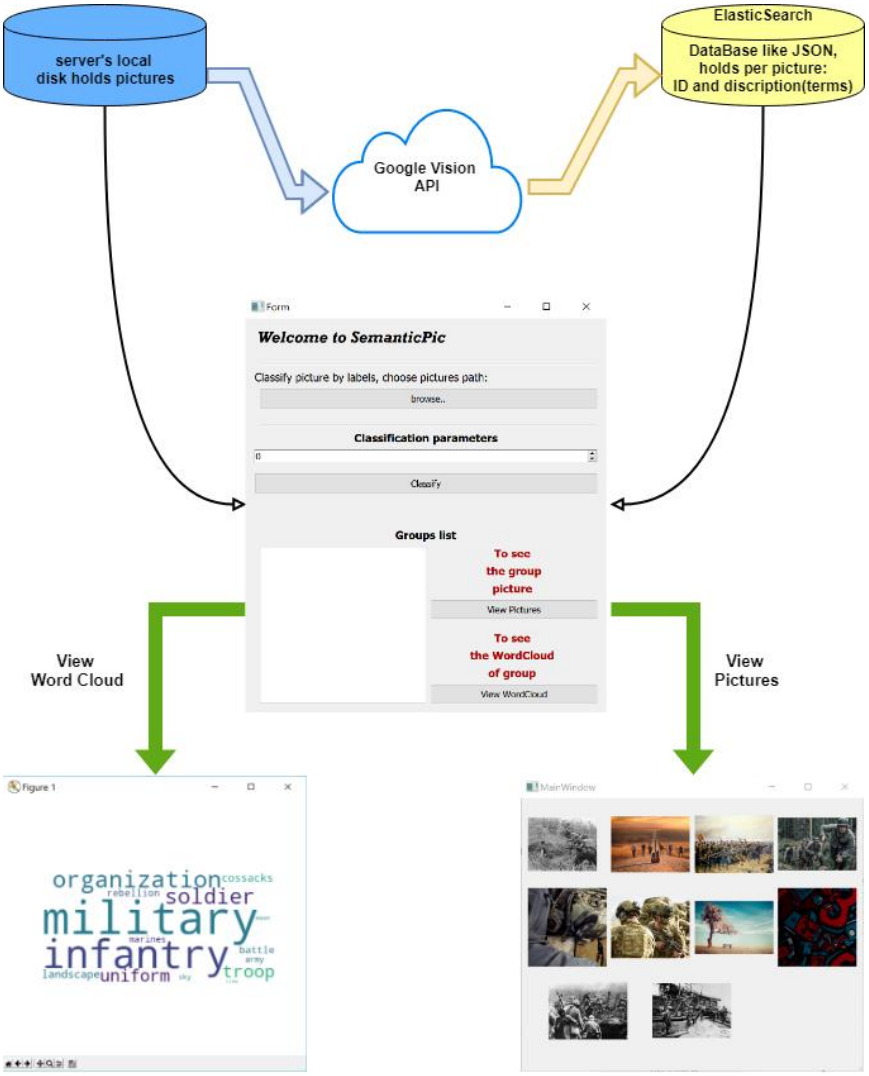
תרשים מס' 2 – אינטראקציית המשתמש עם המערכת

התרשים מתאר את הפעולות שהמשתמש יכול לבצע ואת הגורמים החיצוניים שמעורבים בכל סוג פעולה.



תרשים מס' 3 – ביצוע הפעולות של המערכת

התרשים מתאר את הפעולות שמתבצעת במהלך הרצת המערכת שנדרשות לקבלת התוצאות המדויקות.



Abstract

As part of the studies for B.Sc. in software engineering at Azrieli – college of engineering Jerusalem, during the fourth year of studies I conducted a research final project under the direction of Dr. Miriam Alalouf.

The main objective of the project was to obtain metadata from the database stored in the ElasticSearch database and analyze it and divide the images by content.

There is a great deal of difficulty in analyzing and distributing a vast reservoir of images to images groups according to similar content. Therefore, in this project, instead of analyzing the image, there is an analysis of the words that describe the picture, which makes it possible to better access the images in the database.

There's no easy way to find images with similar content except for manual search until all images shared with the same content are found.

To this end, I built the surgeon's engine using NLP (Natural Language Processing) and machine learning techniques, and word collections that describe an image and identify the words that matter and define the image in the best way.

The tool was installed on the college server that enables remote access and was tested on various types of databases.

Azrieli
College of Engineering
Jerusalem

עזריאלי
מכללה אקדמית להנדסה
ירושלים

Software Engineering Department

Tagging images by the most inclusive category

by

Seif Alden Farhan

Academic Supervisor:
Dr. Miriam Alalouf

Azrieli
College of Engineering
Jerusalem

עזריאלי
מכללה אקדמית להנדסה
ירושלים

Software Engineering Department

Tagging images by the most inclusive category

by

Seif Alden Farhan

July 2019 **Tamuz 5779**