

המחלקה להנדסת תוכנה

פרויקט גמר – תשפ"א

מצא את ההבדלים



Spot The Difference

בשיתוף



מגישות

אילה ברזני
שינדי פרנקל
319042362
207191131

מנחה אקדמי: פרופ' רחל בן אליהו - זהרי	אישור:		תאריך: 21.01.21
אחראי תעשייתי: מר מיכאל שטאל	אישור:		תאריך: 21.01.21
רכז הפרויקטים: ד"ר אסף שפנייר	אישור:		תאריך:

תוכן העניינים:

3	מידע	.1
4	מבוא	.2
5	תיאור הבעיה	.3
5	3.1 אפיון הבעיה	
5	3.2 דרישות	
5	3.3 אתגרים טכנולוגיים	
6	תיאור הפתרון	.4
6	4.1 סקירה	
6	4.2 תיאור הפתרון	
8	4.3 תיאור המערכת	
9	תכנית בדיקות	.5
9	סיכום ומסקנות	.6
11	נספחים	.7
11	7.1 תרשימים	
13	7.2 ניהול סיכונים	
13	7.3 טבלת דרישות	
14	7.4 תכנון הפרויקט	
14	7.5 ביבליוגרפיה	

מערכות ניהול הפרויקט:

#	מערכת	מיקום
1	מאגר קוד	GitHub
2	יומן	Trello
3	סרטון דו"ח אלפא	Drive

מידע נוסף:

סוג הפרויקט	3א - חברת הייטק
פרויקט מח"ר	כן
פרויקט ממשיך	פרויקט חדש

מילון מונחים:

- נקודות עניין (Feature Points): האזורים האופציונליים לשינוי בתמונה שהתקבלו לאחר הפעלת האלגוריתמים הרלוונטיים.
- מודל: הרשת שנבנה על מנת לשפר את ביצועי האלגוריתם כך שהאזורים הנבחרים לשינוי יהיו אופטימליים.

מבוא

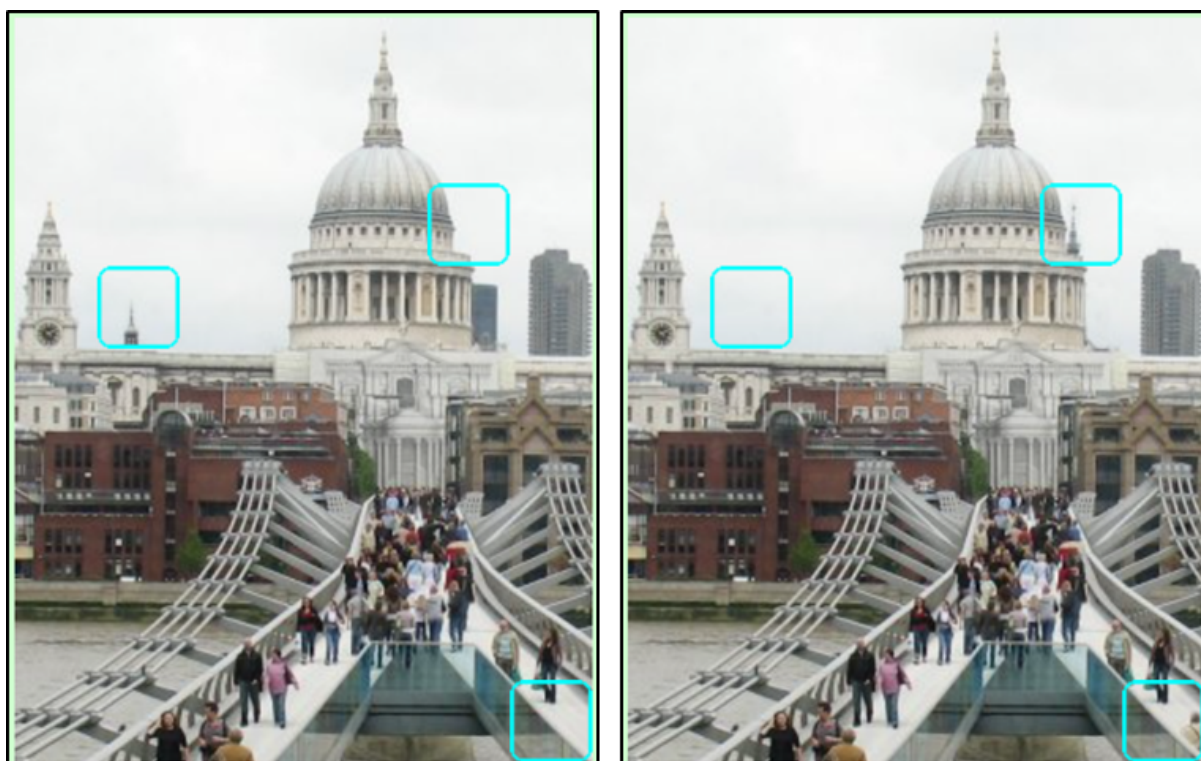
במסגרת הפרויקט נפתח אלגוריתם חכם היוצר באופן דינמי תמונה חדשה על פי תמונת קלט עבור משחק 'מצא את ההבדלים'. מטרת הפיתוח היא לאפשר למשתמש להעלות תמונה שיבחר ולקבל בקלות משחק מוכן.

הפרויקט מתבצע עבור חברה המפתחת משחקים ממוחשבים לחברות ולאנשים פרטיים, פיתוח האלגוריתם יאפשר ליצור כמות בלתי מוגבלת של וריאציות למשחק 'מצא את ההבדלים' ובכך להגדיל את השימוש בפלטפורמה.

האפשרות למשחק זה כיום היא ביצוע השינויים בתמונה מראש באופן ידני באמצעות תוכנות גרפיות, האלגוריתם שנפתח ישנה את התמונה בצורה אוטומטית ע"י מציאת הבדלים אופציונליים בתמונה, ביצוע השינויים ואינטגרציה כוללת עד לקבלת המשחק.

ניגע בתחומים של ראייה ממוחשבת ולמידת מכונה, על מנת למקסם את איכות התמונה המתקבלת ולמזער את השגיאות, כאשר האתגר העיקרי הוא לאמן את האלגוריתם לבחור את השינויים בצורה חכמה, לאחר מכן לבצע אותם בצורה חלקה וליצור תמונת פלט הגיונית.

הדוגמה הבאה מציגה 2 תמונות עם שינויים קלים שבוצעו בעזרת תוכנה גרפית:



תיאור הבעיה

אפיון הבעיה

על מנת ליצור שינויים בתמונה נדרש ידע בתוכנה גרפית, משתמשים שרוצים ליצור את המשחק ללא ידע מוקדם לא יכולים לעשות זאת, כמו"כ יוצרי המשחק חייבים לעשות את השינויים מראש בצורה ידנית ומסורבלת והם אינם יכולים לשחק.

מטרת הפרויקט היא לאפשר ייצור של משחקים באופן אוטומטי ללא צורך בידע מוקדם בגרפיקה של המשתמשים. לקוחות החברה מעוניינים לקבל מגוון של משחקים לשימושים שונים והמשחק שיפותח נותן מענה מתאים בכך שהוא מוריד את הצורך בידע מוקדם ובפעולה ידנית והופך את הכל לאוטומטי.

דרישות

ע"ע טבלת דרישות בנספחים.

אתגרים טכנולוגיים

- זיהוי אזורים מתאימים לשינוי - זיהוי נקודות העניין בתמונה בצורה גנרית שתתאים לכל סוגי הסצנות/האובייקטים שיכולים להתקבל בתמונה, כאשר מתוכם ייבחרו האזורים המתאימים ביותר ליצירת השינוי.
- עיבוד התמונה - הורדת/שינוי אזור מסוים בתמונה בצורה שתראה חלקה לעיניי המשתמש ויחד עם זאת שונה באופן ממשי כך שניתן להצביע על האזור כשינוי.
- איסוף מידע - יצירת סט תמונות אמין עבור המודל כך שהאלגוריתם ידע לבחור את השינויים בצורה אופטימלית. הסט צריך לייצג טווח רחב של סצנות ומגוון של שינויים על מנת להגיע לאחוז דיוק גבוה.
- בחירת מודל מתאים ביותר - נרצה לבנות רשת שתדע לבחור/לבצע את השינויים בצורה הטובה ביותר, לצורך כך נצטרך לחקור את האלגוריתמים השונים ולבחור את המתאים ביותר לדרישות ולתוצאה המתבקשת.
- בחירת פיצ'רים מתאימים ביותר - על מנת לאמן את המודל שיבחר, נרצה לבחור פיצ'רים שיהוו ייצוג הולם לכל סוגי התמונות האפשריות וישקפו נכונה את התוצאה המתבקשת.
- ביצועים - האלגוריתם צריך לספק את התוצאה המתבקשת במהירות סבירה.
- אינטגרציה - הטמעת האלגוריתם בסביבת הלקוח.

תיאור הפתרון

סקירה

בשלב הראשון נעסוק בעיקר בפירוק התמונה לאובייקטים, בזיהויים ובפיתוח היכולת לבצע מניפולציות על התמונה (שלב ה Computer Vision). מטרת השלב היא ליצור dataset מגוון של תמונות קלט עם תמונת הפלט שהאלגוריתם הוציא.

בשלב שני (שלב ה Deep Learning), לאחר שיש בידינו היכולת לבצע שינוי בתמונה ו-data לאימון הרשת, נלמד את המערכת לבחור מיקום/שינוי בצורה הטובה ביותר לפי פרמטרים שהוגדרו ולפי התוצאה שהתקבלה מהמודל.

תיאור הפתרון

שלב א: Computer Vision

בשלב זה נגדיר את הצעדים הבאים:

א. זיהוי והגדרה:

- זיהוי נקודות עניין (feature points) בעזרת GaussianBlur ואלגוריתם Canny.
- סינון הנקודות שהתקבלו לפי פרמטרים שונים כגון: גודל, חפיפת נקודות עניין, שונות נמוכה בצבע בקצוות המלבן, השוואה לנקודות עניין שהתקבלו מאותה תמונה על ידי הפעלת סף אחר ועוד.
- זיהוי תווי פנים על מנת לבצע עליהם פעולות שונות כגון: הורדת/עצימת עין וכן כדי לפעול באזור זה בזיהוי יתירה (למשל סינון נקודות העניין החופפות לאזור הפנים).
- השוואה לתמונת הסגמנטציה על מנת להחליט אם נקודת העניין רלוונטית.

ב. ביצוע:

- בחירת מספר שינויים על פי פרמטרים כגון: רמת המשחק הנדרשת, גודל השינוי הרצוי, אזורים שונים בתמונה ועוד.
- מחיקת אובייקט - על פי תמונת הסגמנטציה נוריד מהתמונה המקורית את האזור המבוקש (הצבע המנוגד לצבע השולט באזור המסגרת). אפשרות נוספת: שימוש באלגוריתם סגמנטציה בעזרת Deep Learning.
- שינוי אובייקט - נבצע את השינוי על פי הגדרה מראש (למשל עצימת עין).
- הוספת פריט - נפזר את השינויים ב 2 תמונות פלט ובכך נדמה הוספת פריט על ידי מחיקתו מתמונת הפלט השניה.
- החזרת התמונות, וקטור קואורדינטות השינויים ורדיוסם.

כלים ועזרים נוספים:

- ספריות פייתון ייעודיות: OpenCV, NumPy
- שימוש באלגוריתמים כגון: Gaussian, Canny

חלוקה למודולים:

- דיאגרמת הרצף הבאה מתארת את התהליך שהאלגוריתם מבצע בשלב זה, החל מקבלת תמונת הקלט מהמשתמש ועד להוצאת תמונת פלט עם שינויים עבור המשחק.

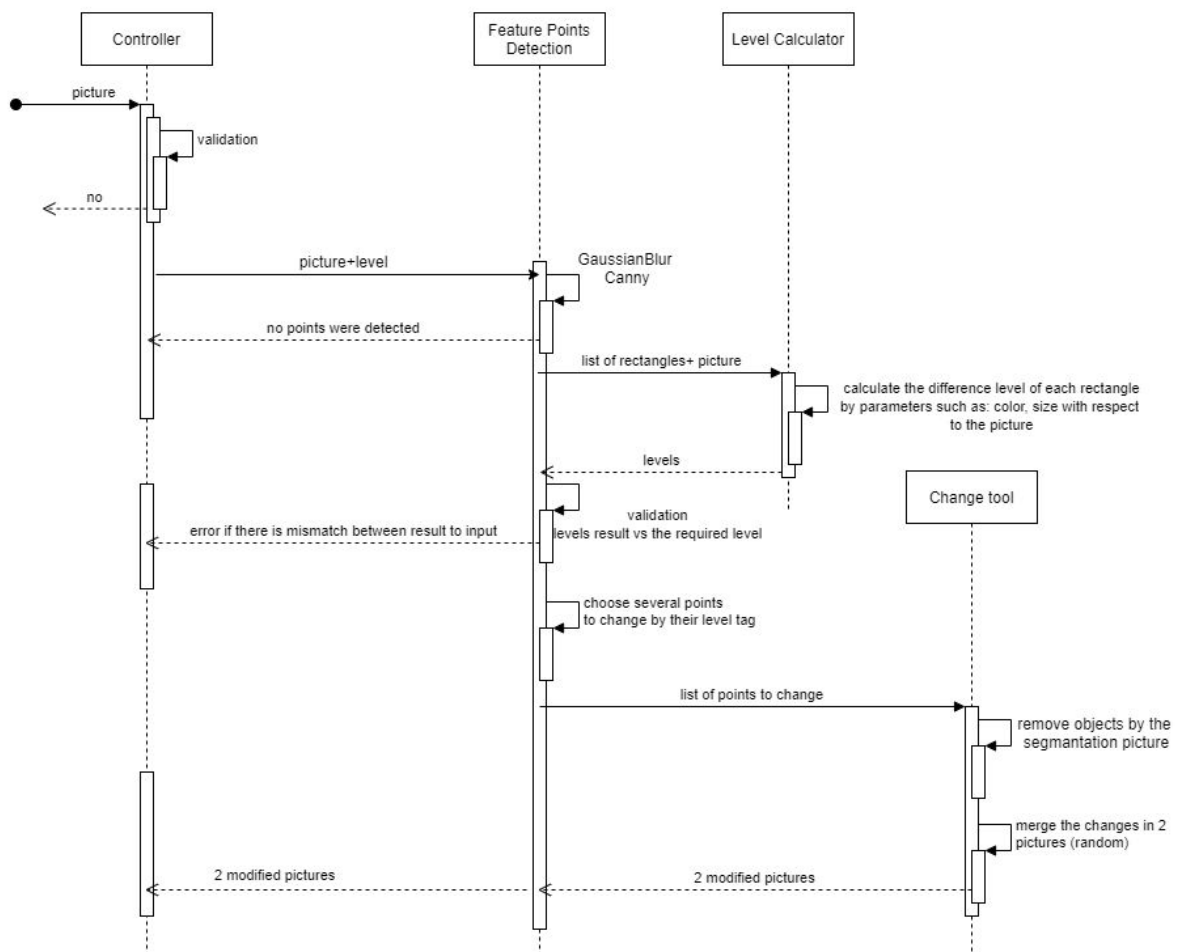


Figure 1

שלב ב: Machine Learning

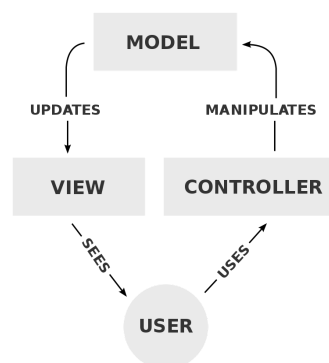
נחקר אלגוריתמים אופציונליים של Machine Learning. נבדוק שני סוגים של רשתות: רשתות יוצרות (Generative Adversarial Networks) - על מנת לבדוק אפשרות של יצירת התמונה ע"י הרשת, ורשתות מסווגות - סיווג השינויים שיצרנו האם הם טובים או לא ובכך לבדוק עבור תמונה חדשה והשינויים האופציונליים עליה, האם הם טובים או לא.

אסטרטגיה:

- בחירת מודל טוב ביותר מבין האפשרויות שייחקרו.
- בחירת פיצ'רים מתאימים ביותר עבור המודל שייבחר.
- הכנת ה-data וסיווג השינויים שהתקבלו (במידה ויהיה צורך).
- אימון המודל על ה-data וקבלת משקלות אופטימליות.
- אנליזה של התוצאות שהתקבלו עבור שיפורים אפשריים.

תיאור המערכת

המערכת הכוללת מבוססת ארכיטקטורת MVC, כאשר הפרויקט מתמקד בחלק הלוגי.



[Figure 1](#) דיאגרמת רצף

[Figure 2](#) דיאגרמת תרחישי שימוש

[Figure 3](#) דיאגרמת מחלקות

תכנית בדיקות

בדיקות יחידה

- בדיקת כל מחלקה בנפרד על פי הלוגיקה שאותה היא אמורה לממש.
- בדיקת פונקציות ספציפיות שהתוצאה עבורן ניתנת לבחינה (כגון Filtering)

בדיקות מערכת

- בדיקת Flow מקצה לקצה ע"י הכנסת תמונה שתוצאתה ידועה מראש.
- בדיקת תפקוד האלגוריתם בזמן אמת (מבחינת ביצועים וכיוצ"ב).
- בדיקת המודל ע"י יצירת Testing Set.

בדיקות אינטגרציה

- בדיקת פעולת המערכת לאחר הטמעת האלגוריתם.
- בדיקת ביצועים של המערכת השלמה.
- בדיקת התממשקות Client - Server.

סיכום ומסקנות

סיכום:

עד כה עסקנו בניתוח התמונה; מצאנו נקודות עניין בעזרת GaussianBlur, Canny ועוד. לאחר מכן פילטרנו את הנקודות שהתקבלו לפי פרמטרים שונים כגון: גודל, חפיפה, חפיפה לאובייקטים שהוגדרו מראש והשוואה לתמונות עם Threshold שונה. בנוסף, פסלנו נקודות עניין שהצבע שלהן היה זהה ברוב שטחו וזאת על פי תמונת הסגמנטציה.

על מנת לבחור את השינויים שיבוצעו, מוגרלים מספר שינויים מתוך וקטור השינויים שבידינו עם התחשבות במיקומי המלבנים ובשונות הפיקסלים ביחס לסביבה. הורדת אובייקטים שנבחרו תתבצע על פי תמונת הסגמנטציה, כאשר פיקסלים מאותו צבע יורדו מהתמונה המקורית וימולאו לפי ממוצע הרקע מסביב.

כרגע, בחרנו לבצע שינוי יחיד על מנת למקד את הגדרת ותכנון הפתרון, כמו כן כדי לאפשר בהמשך ביצוע של מספר שינויים בצורה רקורסיבית או איטרטיבית, כאשר נשלח את התמונה לביצוע שינוי יחיד מספר פעמים (עם הגבלות לגבי האזורים שכבר בוצע בהם שינוי).

מסקנות:

- זיהוי אובייקטים כפי שחשבנו בתחילה אינו פתרון מספיק טוב עבור ביצוע שינויים, יש לשלב את האפשרות הזו בצורה מוגבלת עם האלגוריתם שהגדרנו למציאת נקודות עניין.
- פילטור נקודות העניין על פי תמונת הסגמנטציה נותן תוצאה טובה עבור נקודות עניין שאינן רלוונטיות לביצוע שינוי.
- נמשיך לחקור אפשרות של שימוש באלגוריתמי GAN ובמידה ונמצא כי זה לא מתאים לצרכים שלנו, נשתמש ברשת מסווגת כך שעבור תמונת קלט ושינוי אפשרי, הרשת תדע לזהות שינוי זה כטוב או לא טוב.
- בשלב ראשון נדע לבצע שינוי אחד (האופטימלי ביותר) בתמונה כאשר יושם דגש על גנריות שתאפשר הרחבה עתידית.
- יש לחקור מהי כמות התמונות הנדרשת עבור ה Data Set המתאים.
- הגדרת הפרויקט וחקירת הנושאים הרלוונטיים לקחה יותר זמן מכפי שחשבנו.

נספחים

תרשימים

דיאגרמת Use Case:

התרשים הבא מתאר את שני השלבים העיקריים בפיתוח, כאשר החלק הראשון (Stage I) מציג את תרחיש השימוש בשלב פיתוח האלגוריתם והחלק השני (Stage II) מציג את תרחיש השימוש במערכת (מה שיתבצע מרגע שהמשתמש יכניס תמונה ועד שיקבל את תמונת הפלט האופטימלית חזרה).

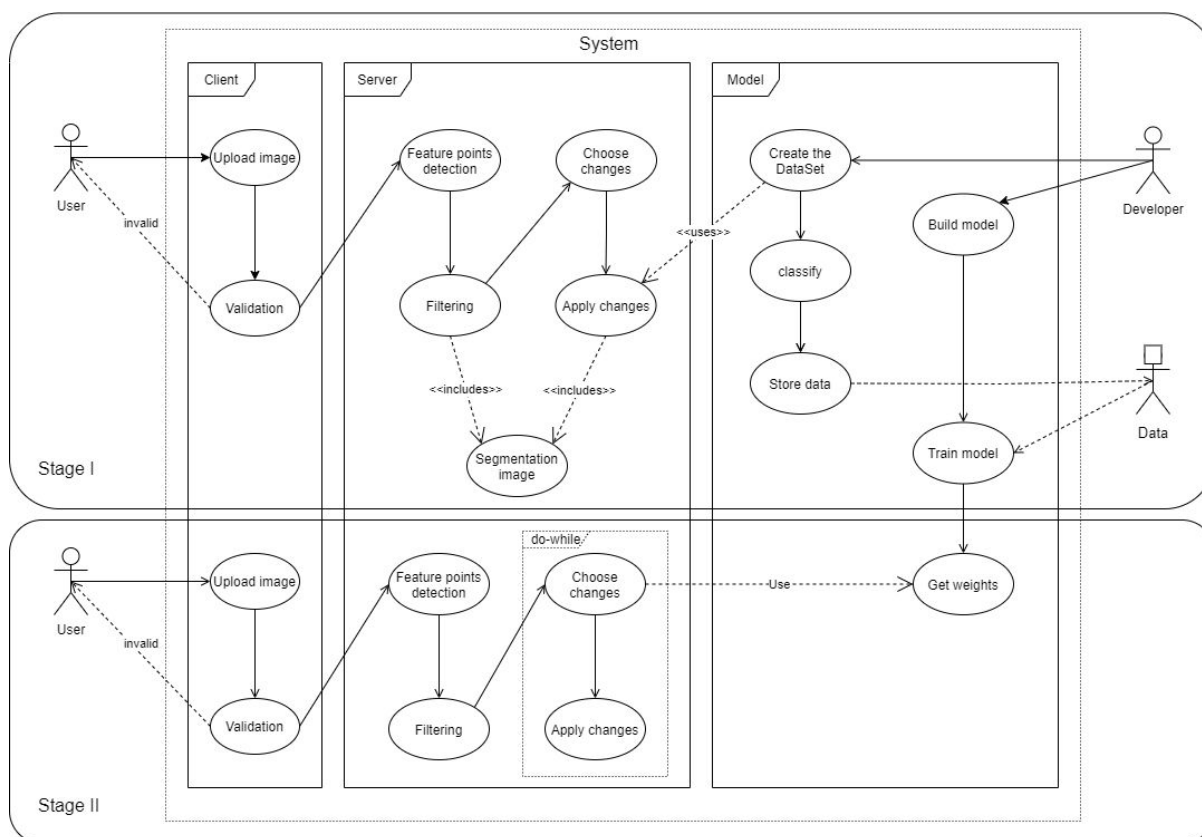


Figure 2

דיאגרמת מחלקות:

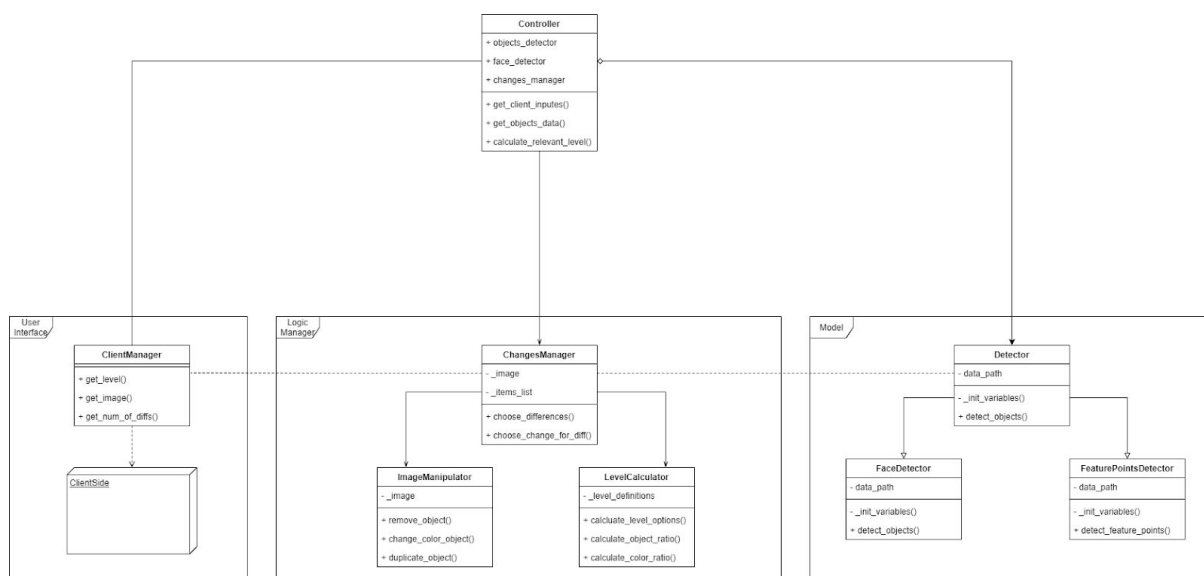


Figure 3

ניהול סיכונים

#	הסיכון	רמת סיכון	השפעה ומענה אפשרי
1	תמונת הסגמנטציה לא נותנת אינדיקציה מספקת לביצוע השינויים.	נמוכה	השפעה אפשרית: זמן ריצת המערכת וביצוע שינויים לא רלוונטים. מענה אפשרי: חקירת מספר אלגוריתמי סגמנטציה ובחירת הטוב מביניהם.
2	פילטור נקודות עניין בתמונה שיכולות להוות שינוי טוב.	נמוכה	השפעה: פספוס שינויים אופציונליים. כל עוד מצאנו נקודות עניין אחרות, ההשפעה תהיה שיתבצעו שינויים פחות טובים על התמונה. מענה אפשרי: ביצוע הפילטור ע"י כמה פרמטרים מתחומים שונים.
3	אלגוריתם למידת המכונה לא עומד בדרישות המערכת.	גבוהה	לא הצלחנו לבנות מודל שיודע לסווג/לבחור בצורה חלקה ונכונה את השינויים בתמונה. השפעה: בזבוז זמן וירידה באיכות האלגוריתם. מענה אפשרי: יש לחקור את כל האפשרויות עם מומחה ולהכין תכנית מגירה.
4	שינוי אסטרטגיית הפתרון בשלב מאוחר.	גבוהה	השפעה אפשרית: זמן יקר שהולך לאיבוד ואי עמידה ביעדים. מענה אפשרי: יש להשקיע זמן בתכנון לפני התחלת הביצוע.
5	הטמעת הקוד בסביבת הלקוח נכשלה.	בינונית	השפעה: יהיה צורך בהתאמת הטכנולוגיה בצד שלנו או בצד הלקוח. מענה אפשרי: תיאום מראש עם החברה ובדיקת ההטמעה כבר בשלב ראשוני.

טבלת דרישות

#	הדרישה	רמת חומרה	תיאור
1	קבלת תמונה מהמשתמש.	גבוהה	התמונה תשלח לשרת לאחר בדיקת תקינותה.
2	מציאת נקודות עניין בתמונה.	גבוהה	על פי אלגוריתמים שהוגדרו מראש, יימצאו אזורים אופציונליים לשינוי בתמונה.
3	בחירת האזורים הטובים ביותר לשינוי.	גבוהה	על פי אלגוריתמים שהוגדרו מראש, נסנן את נקודות העניין שהתקבלו ונבחר את הטובות.

4	בחירת רמת הקושי במשחק.	נמוכה	המשתמש יוכל לבחור באיזו רמת קושי השינויים יהיו. יפותח בשלב מתקדם יותר.
5	השינויים יפוזרו בשתי תמונות פלט נפרדות.	בינונית	על מנת להקשות על זיהוי ההבדלים השינויים יתבצעו על שתי תמונות נפרדות.
6	האלגוריתם יחזיר את וקטור השינויים ורדיוסם.	גבוהה	על מנת לבדוק שהמשתמש זיהה את השינוי הנכון, נצטרך לדעת את האזור של כל שינוי.

תכנון הפרויקט ולוח זמנים

[Spot The Difference - Trello](#)

ביבליוגרפיה

- [1] GaussianBlur mechanism: https://en.wikipedia.org/wiki/Gaussian_blur
- [2] Canny algorithm: https://en.wikipedia.org/wiki/Canny_edge_detector
- [3] OpenCV library documentation: <https://docs.opencv.org/master/>
- [4] GAN: https://en.wikipedia.org/wiki/Generative_adversarial_network
- [5] CNN: https://en.wikipedia.org/wiki/Convolutional_neural_network
- [6] Image segmentation: https://en.wikipedia.org/wiki/Image_segmentation
- [7] TensorFlow library: <https://www.tensorflow.org/>
- [8] pyimagesearch - DL and CV tutorials: <https://www.pyimagesearch.com/>
- [9] Image segmentation using Deep Learning by Er Nupur: <https://medium.com/swlh/image-segmentation-using-deep-learning-a-survey-e37e0f0a1489>