

Ben-Gurion University of the Negev
Faculty of Engineering Science
Dept. of Electrical and Computer Engineering

Digital Image Processing Final Project

Project Name	Jigsaw Puzzle Solver
Students	Ohad Shapira Orel Mantel
Supervisors	Dr. Tamar Riklin-Raviv
Submitting date	05.03.21

תוכן עניינים

3	תקציר
3	תיאור הבעיה
3	תיאור המערכת
3	אתגרים
4	הנחות
5	שימוש בתכנות מונחה עצמים
6	אלגוריתמים:
6	הפרדת החלקים מהרקע (סגמנטציה)
7	הפרדת החלקים
8	זיהוי פינות
9	סיווג הפאות
10	"טיול על השפה" - הוצאת וקטור הקואורדינטות של כל שפה
11	הוצאת רצועות צבע
12	פונקציית מחיר לפי צבע
13	פונקציית מחיר לפי צורה
14	עץ בינארי לפתרון המסגרת
15	פתרון החלקים הפנימיים
15	חיבור החלקים
17	אנליזה כמותית- אלגוריתם פתרון המסגרת
20	מסקנות
20	סיכום
21	נספח א' - פתרון פאזל נוסף - צ'ובקה

תקציר

בפרויקט זה מימשנו אלגוריתם לפתרון פאזלים מסוג Jigsaw Puzzle, במסמך זה נציג את האלגוריתמים שהשתמשנו בהם ואת כיווני החשיבה שהובילו לפתרון שאותו מימשנו. בנוסף, נציג את פונקציות המחיר שהשתמשנו בהם ואת הפתרון המבוסס על תכנות מונחה עצמים על מנת לפשט את הבעיה ולהגיע לפתרון האופטימלי. נציג אנליזה מפורטת של דרך הפתרון שחישבנו עד הפתרון הסופי אותו מימשנו.

נציין כי בעיה זו ידועה ומפורסמת, נעשו בנושא מחקרים רבים ואף יש כנסים שמתעסקים בפתרון פאזלים.

תיאור הבעיה

פתירת פאזלים הינה משימה מאתגרת גם לבני אדם, אשר בשל אופיה היא מצריכה אלגוריתמים מבוססי עיבוד תמונה על מנת לבצע אותה בצורה ממוחשבת.

קושי הפאזל תלוי במספר פרמטרים וביניהם גודל הפאזל, הגוונים והתבניות המוצגות עליו וכן פרמטרים נוספים שנדבר עליהם בהמשך.

לכל אדם יש דרך מועדפת לפתרון פאזל ולכן היינו צריכים לחשוב על אלגוריתם כללי לפתרון אופטימלי על מנת להשיג מינימום שגיאות ושיהיה יחסית מהיר.

תיאור המערכת

כל החלקים מפוזרים על רקע אחיד. תאורה טובה ואחידה על כל החלקים כך שאין החזרת אור מהחלקים.

המערכת תקבל תמונה יחידה כל החלקים, צילום התמונה יעשה מלמעלה כך שהחלקים באותם יחסי גודל.

כדאי לכוון את התאורה כך שהחלקים לא יטילו צל על הרקע.

אתגרים

במהלך שלבי הפיתוח הראשונים שביצענו לפרויקט, נתקלנו במספר אתגרים אשר גרמו לנו מספר פעמים לשנות את כיוון הפיתוח או את דרך החשיבה לקראת הפתרון המיוחל, שיתפנו את סגל הקורס בחלק מאתגרים אלו והם עזרו לנו להתגבר עליהם, אפרט חלק מאתגרים אלו:

- סגמנטציה - פאזלים, מטבעם הם צבעוניים בעלי גוונים שונים, ולכן כאשר ניסינו מספר פעמים לבצע סגמנטציה על מנת להפריד את החלקים מהרקע, נתקלנו בבעיות שכן עבור מודלי GMM שיצרנו ועבודה עם סכמות צבעים קיבלנו כי חלקים נרחבים מהחלקים שלנו משתלבים עם הרקע ו'נעלמים' כאשר ניסינו לבצע את ההפרדה של החלקים.
- מציאת הפינות של החלקים - על מנת שנוכל ליישר את החלקים ע"י סיבוב רצינו לקבל עבור כל חלק את מיקומי הפינות הקיצוניות שלו. היות והחלקים בעלי צורה מורכבת, אחד האתגרים המשמעותיים היה איך להגדיר ולמצוא את ארבעת הפינות של כל חלק.
- חיבור חלקים עם פונקצית מחיר יחידה -
 - מבוססת צורה - בשלב הראשון, ניסינו לפתור את הפאזל אך ורק עם אלגוריתם מבוסס צורה, אך לא הצלחנו להגיע לרמת דיוק שכן חלקים רבים (בייחוד במסגרת) דומים מאוד לחיבורים אחרים ולכן האלגוריתם לא היה מספיק מדויק על מנת להשיג חיבור מושלם של החלקים.
 - מבוססת צבע - ניסינו גם לייצר אלגוריתם הפותר את הפאזלים באמצעות רצועות צבעים על פאות החלקים, אך שמנו לב כי לפעמים קשה (גם בצורה ידנית) לדעת מה החיבור המתאים, לדוגמה:



לכן, על מנת לפתור את הבעיה, שילבנו את פונקציות המחיר לפונקציה אחת המחשבת את החלק האופטימלי על בסיס צורה וצבע.

הנחות

- הרקע צריך להיות אחיד ובצבע שונה מהצבעים של החלקים.
- החלקים הם בצורה סטנדרטית, כלומר קרובים לצורת ריבוע, בעלי 4 פינות והפאזל המחובר מלבני.
- עקב ההנחה שצורת החלק והצבעים בכל חלקי הפאזל חיוניים לפתרון מדויק, בתמונת הקלט לא יהיו חפיפות או כל מגע בין חלקי הפאזל, אלא יהיו מבודדים לחלוטין אחד מהשני.
- תמונת החלקים צריכה להילקח בתנאי תאורה טובים על מנת שצבעי החלקים המקוריים לא ישתנו בצורה שתפגע בפתרון.

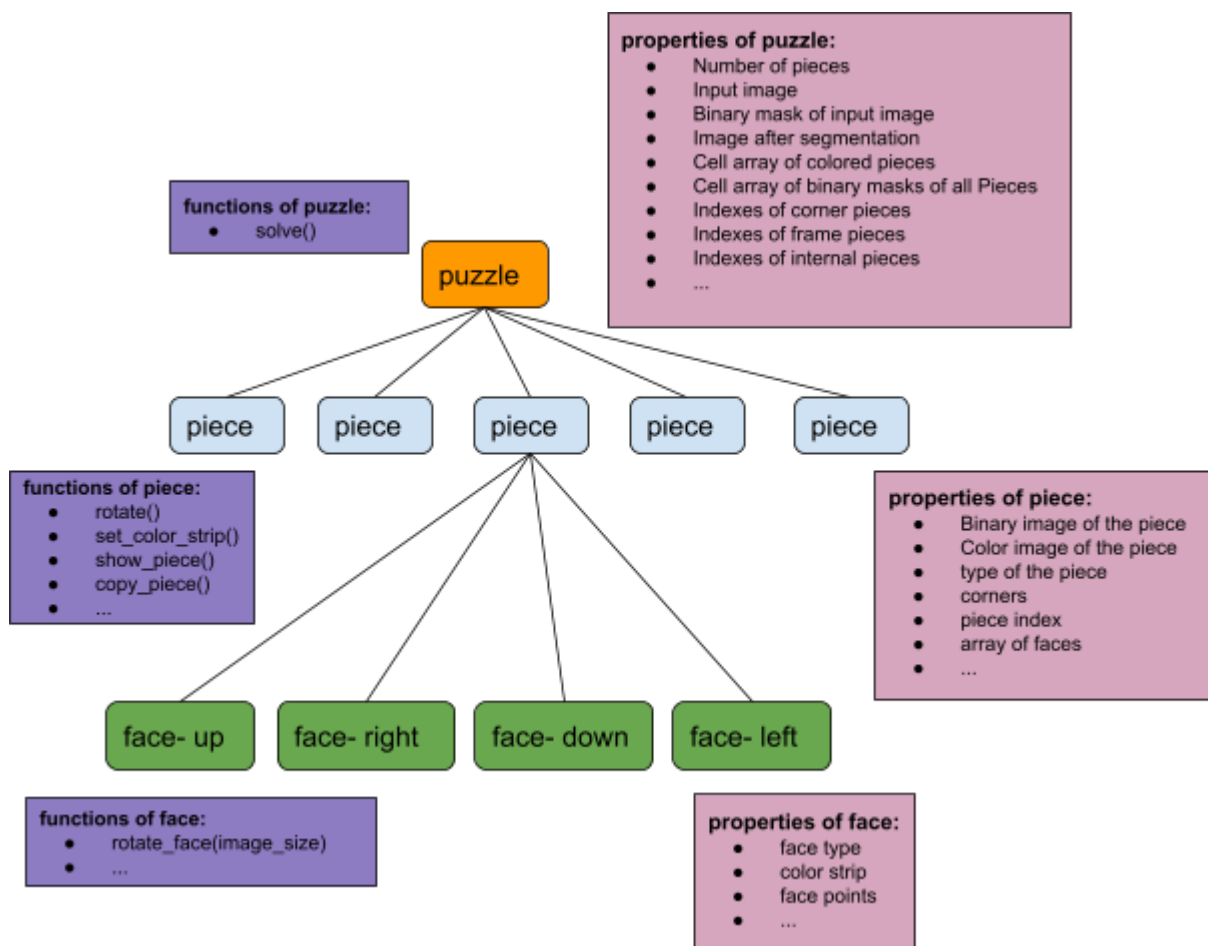
שימוש בתכנות מונחה עצמים

בפרויקט עשינו שימוש בתכנות מונחה עצמים על מנת להקל על העבודה ופישוט האלגוריתמים.

מימשנו 3 מחלקות עיקריות שאיתם אנו מבצעים את הניתוחים ואת העבודה לכיוון הפתרון:

- Face - אובייקט המכיל נתונים על כל פאה בכל חלק- סוג הפאה, רצועת צבע, נקודות שפה.
- Piece - אובייקט המכיל את כל הנתונים על כל חלק, לדוגמה, את ה-Mask, התמונה המוטבעת על החלק, פינות החלק, אינדקס לשימוש עתידי וגם מכיל בתוכו ארבע אובייקטים מסוג Face שהם ארבעת הפאות של החלק, מסודרים בסדר מסוים. בנוסף, מכיל עוד מספר ערכים אשר אנו משתמשים בהם לאורך הפתרון. במחלקה קיימות מספר פונקציות חשובות המבצעות פעולות מסוימות על החלק כמו סיבוב, כך שכל הערכים יהיו עדיין מיוצגים לפי הסדר הנכון.
- Puzzle - מחלקת הבסיס שלנו אשר מכילה בנאי אשר מקבל את התמונה שצולמה ומבצעת הפרדה וניתוח של כל החלקים, ופונקציה נוספת אשר מחזירה את הפתרון המלא של הפאזל.

תרשים המציג את ההיררכיה:



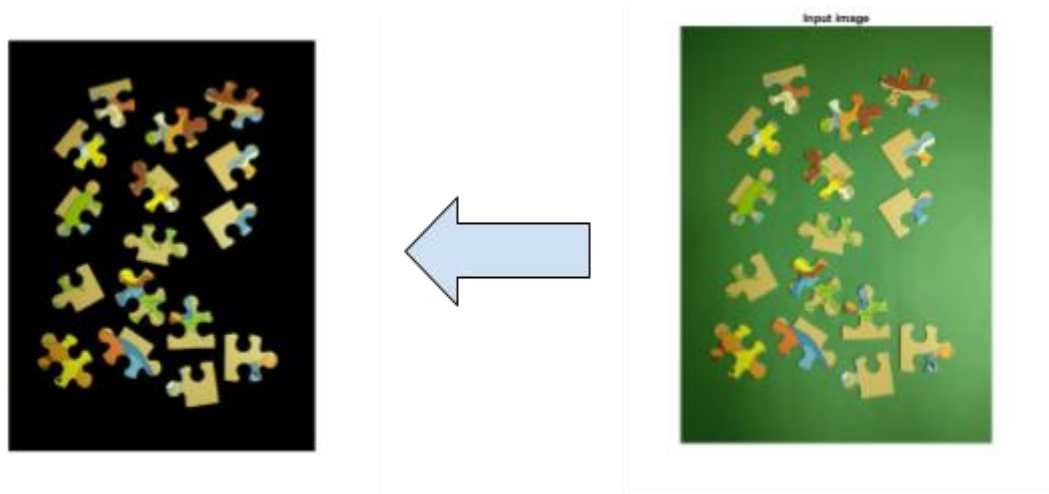
אלגוריתמים:

הפרדת החלקים מהרקע (סגמנטציה)

זהו השלב הראשון באלגוריתם שבו מפרידים את החלקים מהרקע. שלב זה קריטי להצלחת השלבים הבאים היות ובמידה ולא נצליח להפריד את הרקע בצורה טובה, זה יכול לפגוע בזיהוי פינות ואף בזיהוי החלקים. בחרנו לממש את השלב הזה ע"י מודל GMM שמתאים לפאזל המסוים שאנו עובדים איתו. בנוסף, המודל מתאים לרקע ירוק שבחרנו כך שהוא לא מופיע בצבעי הפאזל. אימנו את המודל מראש על כמה תמונות בתנאי תאורה מעט שונים ובזוויות צילום מעט שונות. תמונות האימון שלפיהם הגדרנו את מודל ה-GMM:



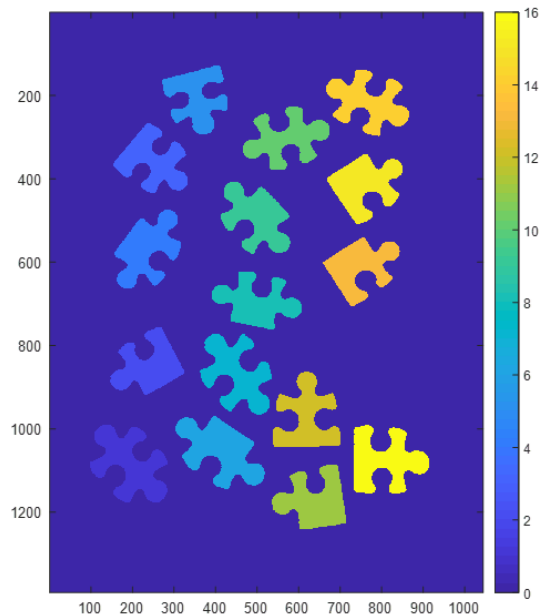
את האימון הרצנו מראש כדי לשפר את זמן הריצה וכדי להגיע לתוצאות אחידות בכל ריצה.



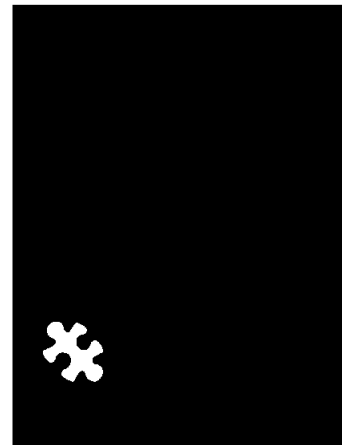
בנוסף, יש לציין כי המודל עובד על הפאזל הספציפי שנבחר. כאשר מחליפים פאזל נדרש לאמן מודל GMM חדש שמתאים לצבעי הפאזל והרקע החדשים.

הפרדת החלקים

לאחר שלב הסגמנטציה, אנחנו מקבלים תמונת mask המכילה את כל החלקים עם מעט רעשים. על מנת להיפטר מרעשים אלה ולקבל תוצאה יותר חלקה (חשוב לשלבים הבאים) אנחנו מעבירים את ה-mask בפילטר על מנת שיתקן את השגיאות. בשלב הבא, אנחנו מסווגים כל רכיב קשירות על ידי אינדקס וכך אנו נוכל לאפיין כל חתיכה בנפרד בעזרת הפונקציה bwlable.



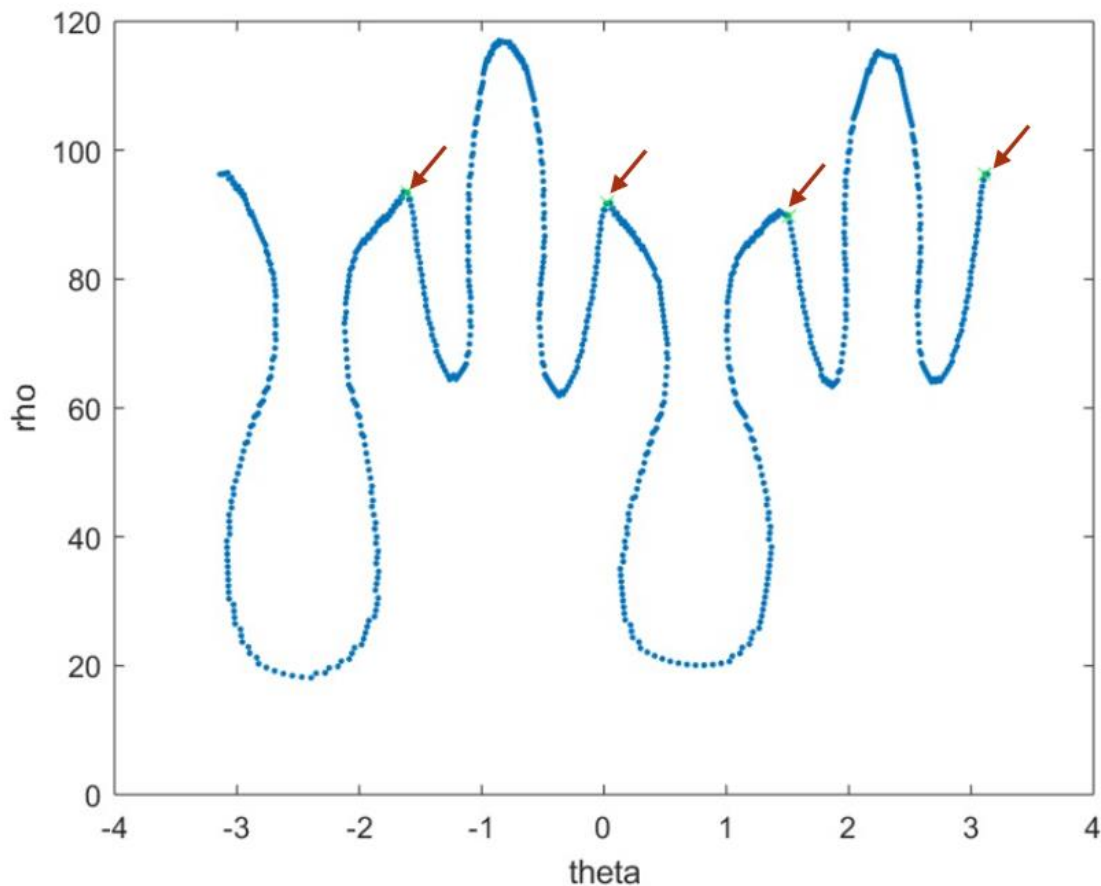
בעזרת השלבים הקודמים, נוכל להפריד כל חתיכה ובנוסף, להכפיל כל mask בתמונה המקורית כדי לקבל כל חלק עם התמונה הצבעונית בנפרד.



שלב נוסף שאנו מבצעים בחלק זה, על מנת שנוכל להשתמש בצורה קלה ונוחה עם החלקים במהלך השוואתם וחיבורם העתידיים, אנחנו מצמצמים את התמונה וחותכים אותה כך שמקבלים את ה-Mask והחתיכה הצבעונית בתמונה קטנה.

זיהוי פינות

על מנת לזהות את הפינות של כל חתיכה בנפרד, תחילה הוצאנו את הקואורדינטות של הפיקסלים המייצגים את השפה של החלק, במישור (x,y) . לאחר מכן העברנו את מערכת הצירים להיות במרכז המסה של השפה. מצב זה פתח בפנינו את האופציה להעביר את הנקודות לקואורדינטות פולריות (מישור (ρ, θ)), ובכך יכולנו לזהות דפוס חוזר של פינות החלקים כאזורים בעלי "שפיצים" בגרף:



גרף זה מחזורי, כפי שה- $edge$ של חלקי הפאזל הוא מחזורי, לכן אין חשיבות לנקודת התחלה וסוף.

על מנת לזהות את החלקים בעלי "השפיץ" המייצגים את הפינות, מצאנו את הנקודות בגרף שמייצגות בצורה החזקה ביותר דפוס מסוים של התנהגות הגרדיאנט של ρ באזורים שלפני ואחרי אותה נקודה. היות והפינות בגרף מתאפיינות בגרדיאנט יחסית גדול וחיובי מצד שמאל שלהן, וגרדיאנט יחסית גדול בערכו ושלילי בסימנו מצד ימין שלהן, כאשר העברנו את הדפוס שאנחנו מחפשים (מיוצג על ידי וקטור של אחדות בצד שמאל ו-(-1) בצד ימין), כפלנו את ערכי הגרדיאנט הסובבים כל אחת מהנקודות בווקטור הזה וקיבלנו מדד שניתן להשוות איתו את כל הנקודות.

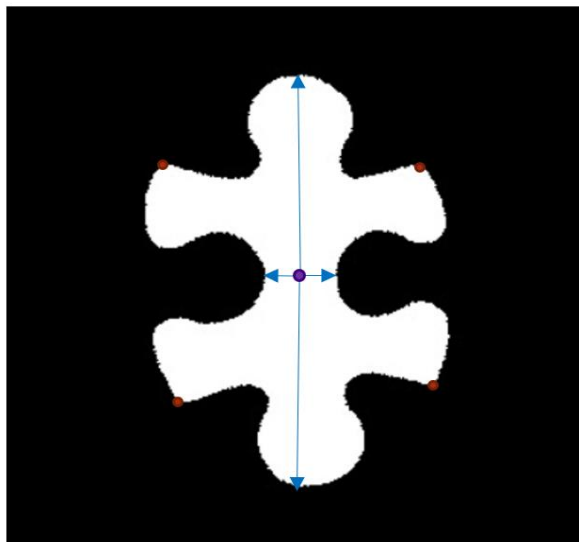
לאחר מכן לקחנו את ארבעת הנקודות בעלות המדד הגבוה ביותר (בניכוי נקודות קרובות אליהן, שיכולות גם הן לקבל ערכים גבוהים), וכך מצאנו את הנקודות המייצגות את הפינות.

כל שנותר הוא להעביר את הנקודות חזרה למישור (x,y) ולזהות אותן על גבי החלק המקורי.

סיווג הפאות

על מנת לסווג את הפאות היינו צריכים למצוא את הפינות, ולאחר מכן לסווג את החלק בהתאם לזווית המקורית שבה הוא צולם כך שהחלק יהיה מיושר, כלומר שהפינות שלו יצרו מלבן בקירוב.

לאחר מכן, סיווג הפאות מתבצע בצורה יחסית פשוטה. חלקי הפאזל הינם חלקים סטנדרטיים, והחלקים שמחברים בין חלק לחלק נמצאים במרכז הפאה בקירוב. ניצול עובדה זו אפשר לנו למצוא את סיווג הפאות בצורה הבאה:

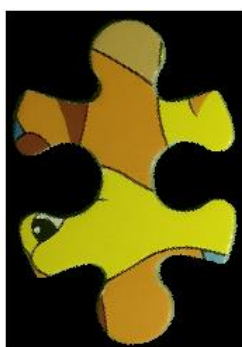


תחילה מצאנו את מרכז המסה של החלק (בסגול), שהוא ממוצע של הקואורדינטות שמייצגות את הפינות (באדום). מכאן בדקנו את המרחק (בפיקסלים) שאנחנו נאלצים לעבור בתוך החלק, עד שאנחנו יוצאים ממנו החוצה ופוגשים את הרקע. היות והחלקים מיושרים, נוכל לבצע את המשימה הזאת בצורה נוחה שכן נוכל לנוע בקווים אופקיים או אנכיים בלבד (בכחול) ממרכז המסה.

בעזרת לוגיקה פשוטה וערך סף שקבענו, אנחנו יודעים לסווג את הפאה לשלוש קטגוריות: פאה שיוצאת החוצה (*head*), נכנסת פנימה (*hole*) או פאה שטוחה (*flat*).

לאחר שאנחנו יודעים לסווג את ארבעת הפאות של כל חלק, נדע גם לסווג את החלק עצמו לאחת משלוש קטגוריות: חלק פנימי (*internal*), חלק מסגרת (*frame*) או חלק פינה (*corner*).

לאחר סיווג הפאות והחלקים, מתבצע סידור החלקים כך שחלקים מהמסגרת (*frame*) יהיו מסובבים כך שהפאה השטוחה תהיה למעלה וחלקי פינה (*corner*) יהיו מסובבים כך שפאה שטוחה אחת תהיה לצד שמאל והפאה השטוחה השנייה תהיה למעלה.



internal



corner



frame

"טיול על השפה" - הוצאת וקטור הקואורדינטות של כל שפה

עבור כל אחת מארבעת הפאות בכל חלק אנחנו מוציאים את המיקומים המדויקים של השפה כאשר הם מסודרים מנקודת פינה אחת לנקודת פינה שנייה. לדוגמא, עבור החלק הבא, נדגים את התהליך על הפאה הימנית:



הנקודה הראשונה היא הפינה הימנית העליונה, נמצא נקודה קרובה לה על השפה באמצעות לוגיקה פשוטה בהנחה שהפינה המקורית מאוד קרובה לשפה או ממש עליה. נקודה זו תהיה נקודת ההתחלה.

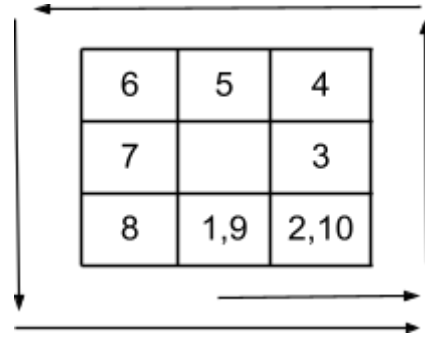


נעשה תהליך דומה גם על נקודת הסוף (הפינה השמאלית התחתונה):



בשלב הבא, הולכים מנקודת ההתחלה לכיוון הנקודת סיום, עם כיוון השעון מסביב לחלק. מכל נקודה, נמצא את הנקודה הבאה לפי התהליך הבא:

1. נוציא מטריצת 33×33 פיקסלים מה-mask מסביב לנקודה.
2. נפרוס כוקטור חדש את הנקודות מסביב לנקודה באופן הבא:



3. נחפש בוקטור החדש את הסדרה $[1,1,0]$.
4. ואז נבחר באיבר השני בסדרה כנקודה הבאה.
(אם לא נמצא, נחפש רק $[1,0]$ וניקח את האיבר הראשון)
5. אם הגענו לנקודת סיום, נעצור, אחרת, נמשיך באותו אופן.
וכך נקבל את המיקומים של כל נקודות השפה עבור הפאה הנתונה:



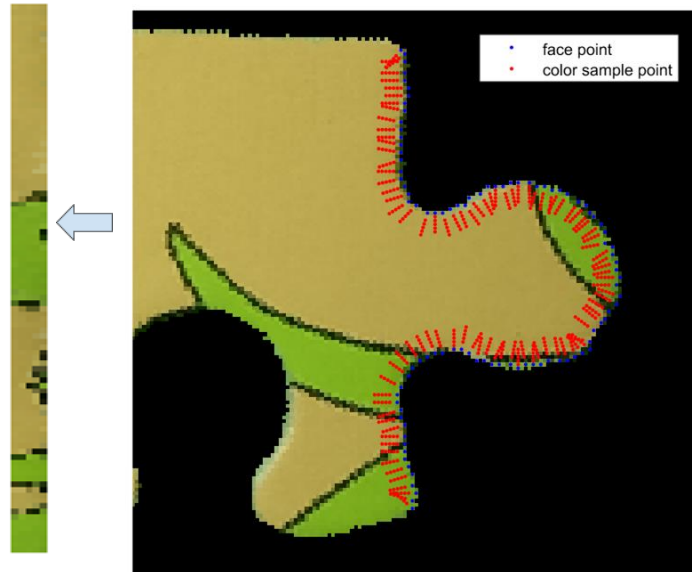
הוצאת רצועות צבע

לצורך ביצוע השוואות צבע בין השפות של שני חלקים, נרצה לדגום את הפיקסלים הקרובים לשפה בכל פאה. הרעיון הוא לדגום 5 נקודות פנימה לתוך החלק על הקו הישר המאונך לשפה. בנוסף, נרצה שעבור כל פאה יהיו לנו מספר שווה של נקודות להשוואה כך שנוכל לבצע השוואה מהירה בין שני חלקים.

נגדיר לדגום 100 נקודות בכל שפה. (מספר זה יכול להשתנות בהתאם לאיכות הצילום)

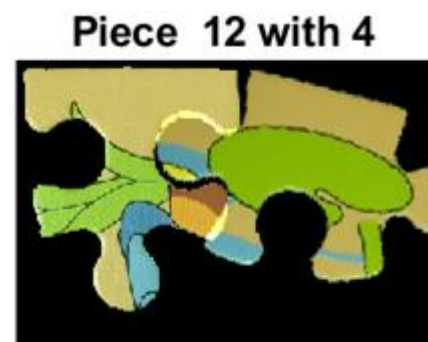
לכל חלק, עבור כל פאה, נעבור על 100 נקודות שפה מתוך הנקודות שיש לנו בקפיצות אחידות בקירוב. בכל נקודה, ניקח מסביבה ריבוע 5×5 מתוך ה-edge של mask ונמצא את הזווית המקורבת של השפה בעזרת הפונקציה regionprops. נרצה להוסיף או להחסיר 90 מעלות כדי להגיע לווקטור שיוצא מנקודת השפה לתוך החלק במאונך. לכן, נבדוק קודם אם צריך להוסיף או להחסיר ואז נדגום נקודות צבע ב 5 רדיוסים שונים מהשפה.

לאחר סיום התהליך לפאה מסוימת, יהיו לנו הדגימות במיקומים הבאים:



פונקציית מחיר לפי צבע

המטרה- פונקציה שתקבל שתי פאות משני חלקים שונים ותחזיר סקלר שמייצג את המחיר שנשלם עבור הבחירה בשני הפאות כחיבור תקין.
נגדיר שהחיבור הוא תמיד אופקי ולכן יש חלק שמאלי וחלק ימני.
עבור החלק השמאלי, דגימות הצבע הפוכות במימד השני כי דגמנו מימין לשמאל ולכן נצטרך להפוך אותם.
עבור החלק הימני, דגימות הצבע הפוכות במימד הראשון כי דגמנו מלמטה למעלה ולכן נהפוך את הסדר.
מתוך ההנחה שהתמונה ברובה רציפה, בכל ערוץ צבע ניקח ממוצע לכל שורה ונחשב שגיאה ריבועית מול השורה המתאימה ברצועה השנייה. נסכום את כל השגיאות ונחלק ב-300 (מספר הנקודות כפול מספר ערוצי הצבע).
לרוב, התוצאה תהיה נמוכה עבור חיבורים מתאימים וגבוהה עבור חיבורים שלא מתאימים.
יש לציין שזה לא תמיד המצב היות ולפעמים יש הבדלים של אור וצל שמגדילים את השגיאה וגם לפעמים האי רציפות בתמונה יוצא בדיוק בחיבור בין החלקים.
לדוגמא, עבור שני חלקים שלא אמורים להתחבר:



המחיר=0.0435

ועבור חלקים שאמורים להתחבר:



המחיר=0.0358

Piece 12 with 9



ניתן לראות שבמקרה זה, המחיר אמנם נמוך יותר אבל היות והכנף בחלק הימני מסתיים בדיוק בחיבור שבין החלקים, המחיר עבור הבחירה הזאת גבוה יחסית.

פונקציית מחיר לפי צורה

האלגוריתם הבסיסי והראשוני שאנו משתמשים בו על מנת לחבר את המסגרת הינו חיבור על פי צורה. אלגוריתם זה מקבל חלק מהמסגרת ומחזיר לנו רשימה של חלקים פוטנציאליים לחיבור המסודרים על פי מחיר החיבור, במהלך כל האלגוריתם אנו משתמשים ב-mask של החלקים מכיוון שהצבע אינו משחק פה תפקיד כלל.

האלגוריתם מקבל חלק מהמסגרת, ומנסה למצוא את החיבור הטוב ביותר לפאה מצד ימין של החלק, לאחר סיווג ראשוני על פי תכונות בסיסיות שאנו מסווגים בשלבים קודמים, האלגוריתם יודע לרוץ על החלקים שיכולים להתחבר לוגית (hole-to head).

לאחר שצמצמנו את רשימת החלקים הפוטנציאליים לחיבור, ובאמצעות 15 נקודות שפה מכל פאה אנו מבצעים לכל אחד מהם טרנספורמציה מסוג 'Non reflective Similarity'. בדומה לטרנספורמציה שאנו עושים בחיבור החלקים בשלב האחרון (ראה עמוד 15).

כפי שציינו, אנו עובדים פה על mask-ים והתמונה מוצגת לשם פישוט והבנה של התהליך

Piece 2 with 3



Piece 2 with 5



Piece 2 with 12



Piece 2 with 16



- לאחר שביצענו טרנספורמציה לחלקים, אנו מבצעים חישוב על ידי שתי פונקציות מחיר:
1. אנו מבצעים and על החלקים לאחר ההזזה שלהם, כך אנו מזהים כמה חתיכה אחת עלתה על השנייה, את פרמטר זה נרצה למזער כמה שאפשר.
 2. אנו מבצעים לחתיכה הראשונית שהאלגוריתם קיבל dilate ולאחר מכן מבצעים AND ובכך אנו מזהים רווחים בין החתיכות, את פרמטר זה נרצה למקסם, שכן עבור החתיכה הטובה ביותר נרצה שלא יהיו רווחים כלל.

באמצעות שתי פונקציות אלה, אנו ערך מספרי המייצג את טיב החיבור בין החתיכות הרלוונטיות לחתיכת הרפרנס. הפונקציה מחזירה מטריצה עם מספרי החלקים הרלוונטיים והמחירים שלהם לפי צורה.

עץ בינארי לפתרון המסגרת

- השלב הראשון בפתרון הפאזל הוא פתרון המסגרת.
- היות וכבר סיווגנו את החלקים ל-3 קבוצות - corner, frame, internal, נוכל להתחיל מאחת הפינות. הרעיון הוא לבנות עץ בינארי כאשר תחילת העץ היא הפינה שאיתה מתחילים ובכל פיצול יש את החלק שמתחבר הכי טוב והחלק השני הכי טוב במובן של פונקציות המחיר לפי צורה ולפי צבע.
- העץ מוגדר כאובייקט עם 3 ערכים:
- ענפים (branches)
 - משקולות (weights)
 - פתרונות (solutions)
- הענפים מייצגים פתרונות אפשריים למסגרת, המשקולות הם וקטורים של המחירים לכל אחד מהחלקים בכל ענף והפתרונות הם cells בגודל הפאזל כשבכל מיקום במסגרת נמצא אובייקט מסוג piece שמייצג את החלק שנבחר למיקום זה, מסובב לפוזיציה שנבחרה.
- בחרנו למלא את העץ בעזרת פונקציה רקורסיבית שמתחילה מהפינה הנתונה וממלאה את העץ בפתרונות אפשריים בלבד כך שמשתמשים בכל החלקים של המסגרת והפינות וגם מגיעים למצב שבו המסגרת שלמה (כך שהתחלנו וסיימנו באותו מיקום).
- שלבי האלגוריתם (באופן כללי):
1. הפעלת פונקציית המחיר לפי צורה על החלק הנוכחי וכל אחד מהחלקים שעדיין לא חוברו.
 2. עבור החלקים האפשריים, הפעלת פונקציית המחיר לפי צבע.
 3. סכימת המחירים לפי צבע וצורה ובחירת 2 החלקים עבורם המחיר הנמוך ביותר.
 4. עבור כל אחד משני החלקים שבחרנו, מכניסים אותו לפתרון במקום הנכון ובסיבוב הנכון, שומרים את המחיר שלו ומפעילים את הפונקציה הרקורסיבית עבור הפתרון הנוכחי.
 5. אם השתמשנו בכל החלקים של המסגרת וגם הגענו למיקום ההתחלתי אז שומרים את הענף, המחירים והפתרון בעץ.
- * במידה והגענו למצב שלא השתמשנו בכל החלקים ואין אף חלק אפשרי לחיבור אז הענף לא נשמר ולא ממשיכים לבדוק אותו.
- ** ניתן להמשיך לחלק השני הטוב ביותר בסעיף 4 רק עד פעמיים עבור כל ענף. כך אנחנו מגבילים את גודל העץ והחישוב מהיר יותר.

בסוף התהליך, מתקבלים מספר פתרונות אפשריים. אנו מבצעים את האלגוריתם הנ"ל 4 פעמים, כאשר בכל פעם מתחילים מפינה אחרת ולבסוף בוחרים בפתרון שעבורו סכום המחירים לכל החלקים הוא הנמוך ביותר. האלגוריתם לא תמיד מחזיר את הפתרון הנכון אך הוא גמיש מאוד ונותן תוצאות יחסית טובות על הדוגמאות שבדקנו.

פתרון החלקים הפנימיים

שלב זה הוא השלב השני של פתרון הפאזל אשר מתבצע לאחר שיש בידינו פתרון של המסגרת. בשלב זה השתמשנו בפונקציית המחיר לפי צבע בלבד בשילוב אלגוריתם שמבטיח שיהיו לפחות 2 פאות שנוכל להשוות אותם עבור כל חלק שננסה להתאים. האלגוריתם מתחיל בפינה השמאלית עליונה (מיקום (2,2) אם נדמה את הפאזל למטריצה) כך שישנן 2 פאות מהמסגרת שמתחברות במיקום זה. אנו מריצים פונקציה שבודקת מי החלקים השכנים באותו מיקום ומחזירה את הפאות שלהם הגובלות עם החלק שאנו מחפשים. לאחר מכן, עבור כל חלק שלא בשימוש עדיין אנחנו בודקים כל אחת מ-4 הפוזיציות (הסיבובים) האפשריות שלו. עבור כל אחת מהפוזיציות, במידה וניתן לחבר את החלק בצורה זו (חיבורים בין head ל- hole בלבד), מחשבים את פונקציית המחיר לפי צבע בין הפאות המתאימות בחלק זה לבין הפאות השכנות שמצאנו. בסוף מתקבלת מטריצת מחירים (בשורות החלקים הנותרים 4 עמודות- אחת לכל פוזיציה) שבה לוקחים את החלק והפוזיציה שמקבל מחיר מינימלי ומתחבר באופן תקין. האלגוריתם עובר שורה שורה עד שכל החלקים מחוברים. יש לציין שבכל הניסיונות שעשינו, כאשר האלגוריתם של המסגרת החזיר תשובה נכונה, גם אלגוריתם זה החזיר תשובה נכונה, מה שמראה שלפחות בפאזלים קטנים, האלגוריתם עובד מצוין.

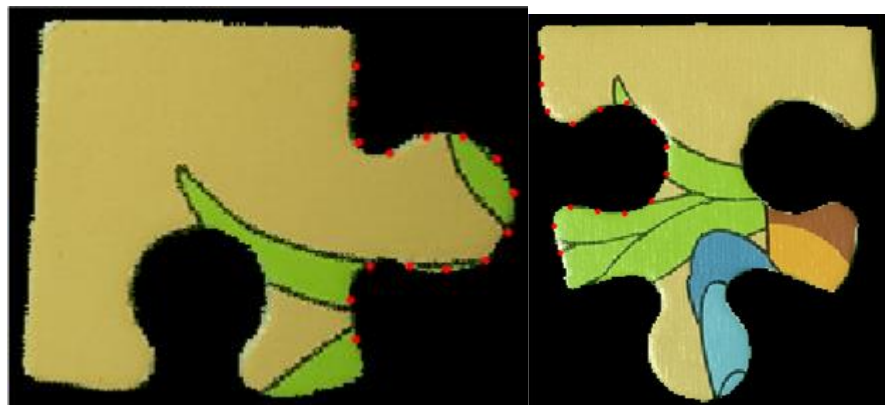
חיבור החלקים

לאחר מציאת ההתאמות האידיאליות לכל אחד מחלקי הפאזל עבור כל פאה, כל שנותר הוא ליצור תמונה המורכבת מחיבור חלקי הפאזל לפי הפתרון שמצאנו.

הרכבת הפאזל נעשית לפי התהליך הבא:

1. עבור הפינה הראשונה (שמאלית למעלה) אנו מעתיקים את החלק בגודלו המקורי ללא טרנספורמציה.
2. עבור שאר החלקים באותה שורה, אנו משתמשים ב-15 נקודות מתוך נקודות השפה ומעבירים אותם למיקומים צמודים לנקודות השפה של החלקים שכבר מורכבים. משתמשים בפונקציה `fitgeotrans` שמוצאת את מטריצת הטרנספורמציה כאשר מחפשים טרנספורמציה אפינית כדי שההתאמה תראה טוב.
3. מבצעים את הטרנספורמציה עבור החלק ומעבירים גם את נקודות השפה שלו לנקודות חדשות לפי אותה טרנספורמציה בעזרת `transformPointsForward` כך שהחלק הבא יוכל להתחבר במיקום הנכון לפי נקודות אלה.
4. נחבר את החלק ע"י חיבור של תמונת הפתרון עם התמונה של החלק המוזז. (במקרים שבהם יש חפיפה בין החלקים הם יהיו בהירים יותר).
5. עבור השורות הבאות בפתרון מבצעים את שלבים 2-4 כך שמוצאים את הטרנספורמציה ע"י שימוש בנקודות שפה בהתאם למספר הפאות השכנות לכל חלק.
6. נחזור על התהליך עבור כל חלק עד אשר נקבל את התמונה המשווה.

דוגמה לנקודות השפה שלפיהם קובעים את הטרנספורמציה:



הצגת הפתרון שלב אחר שלב:



ניתן לראות שקיבלנו תוצאה טובה כך שאין הרבה רווחים בין החלקים והתמונה נראית בבירור.

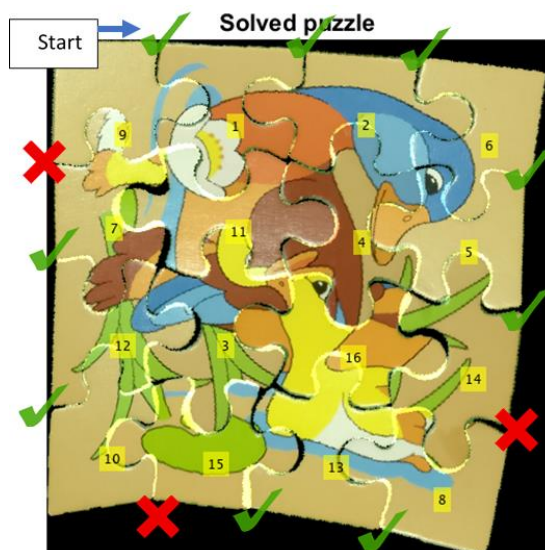
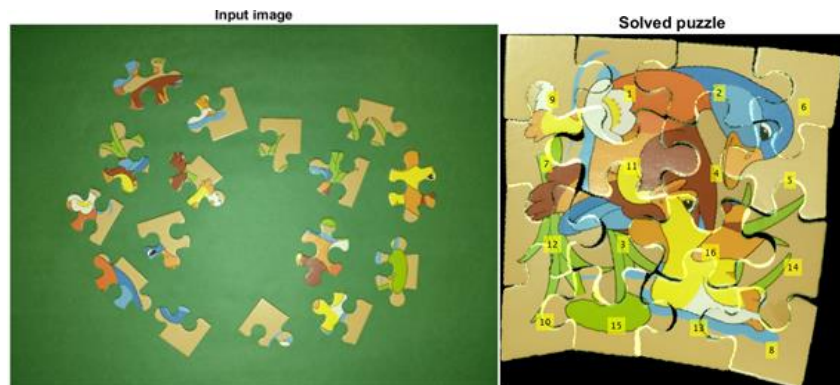


אנליזה כמותית- אלגוריתם פתרון המסגרת

ננתח את שלב פתרון המסגרת שבו אנו מפעילים את שתי פונקציות המחיר. החלקים המקדימים לשלב זה עומדים בפני עצמם ולא דורשים הצלבת נתונים וניתוחים כנגד חלקים אחרים. שלב זה בפרויקט מכיל את מירב החישובים והוא בין השלבים החשובים ביותר לאיכות הפתרון. בשלב זה אנחנו בונים את שלד הפתרון ומנסים להשיג את התוצאה הטובה ביותר על ידי פונקציות המחיר שמימשנו.

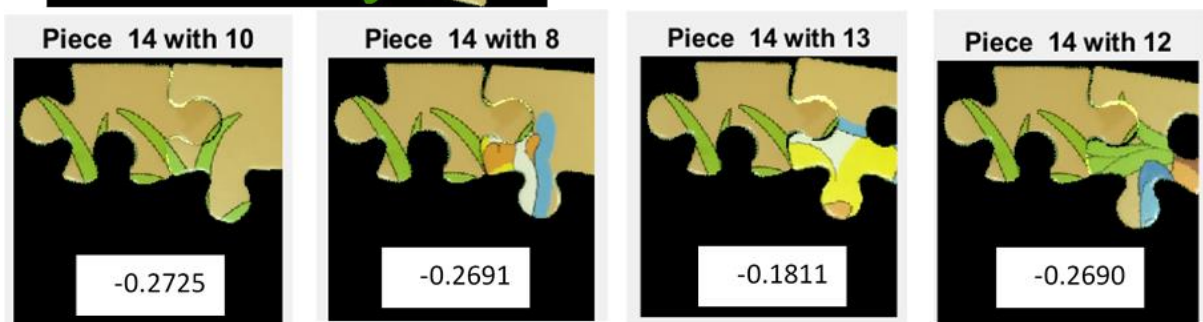
דרך הפתרון שמימשנו מורכבת מכמה שלבים בצורה טורית כאשר בסופם נקבל את החיבור האופטימלי כתלות בפונקציית המחיר שהגדרנו.

מקרה 1 - פתרון שגוי



נשים לב שבמקרה זה, למרות שקיבלנו תוצאה שנראית מבולגנת, קיבלנו $\frac{9}{12}$ הצלחות בבניית המסגרת.

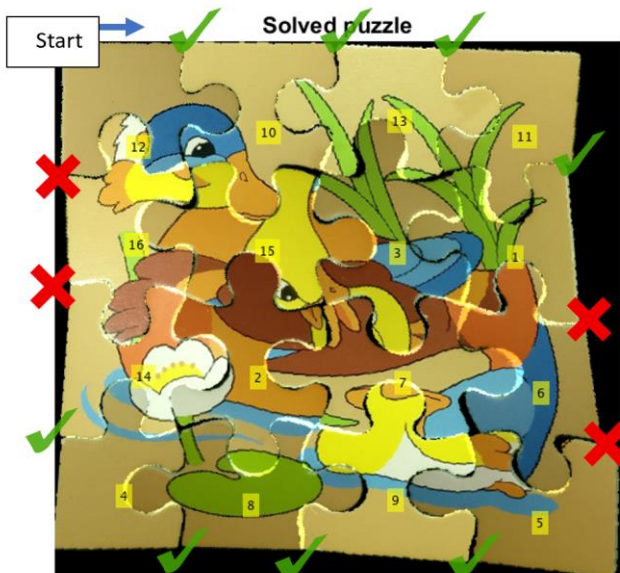
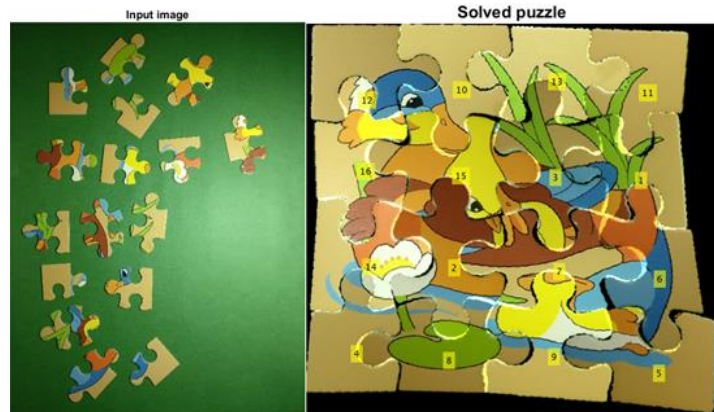
ננסה לנתח את הטעות הראשונה ולהבין למה האלגוריתם טעה. כדי לעשות זאת נגיע לשלב בעץ שבחרים בחלק 14 ונראה מה האפשרויות להמשיך ומה המחיר עבור כל אפשרות:



קיבלנו מצב שהבחירה הראשונה (המינימלית לשלב זה, אשר מתקבל עבור חלק מספר 10) הייתה הנכונה ואילו הבחירה שהאלגוריתם לקח בסוף היא פחות טובה, אך עבור פתרון כל הפאזל, מחיר פתרון המסגרת כאשר נבחר חלק מספר 8 בשלב זה, יהיה מחיר נמוך יותר.

בנוסף, מניתוח האלגוריתם נציין כי בבחירה מוקדמת יותר האלגוריתם בחר דווקא באפשרות הפחות טובה כדי להגיע לתוצאה זו אבל זו הייתה בחירה נכונה. דוגמה זו מראה שגמישות האלגוריתם לא תמיד מביאה את התוצאה הנכונה אך מגדילה את סיכויי ההצלחה.

מקרה 2 - פתרון שגוי






במקרה זה, קיבלנו $\frac{8}{12}$ הצלחות בבניית המסגרת.

הטעות הראשונה היא בבחירת חלק מספר 6 כאשר החלק הקודם לו היה חלק מספר 1.

במקרה זה, כפי שהראנו בהסבר על פונקציית המחיר לפי צבע, מתקבל מחיר גבוה יחסית עבור החלק המתאים מכיוון שהצבעים מתחלפים בדיוק באזור החיבור בין החלקים.

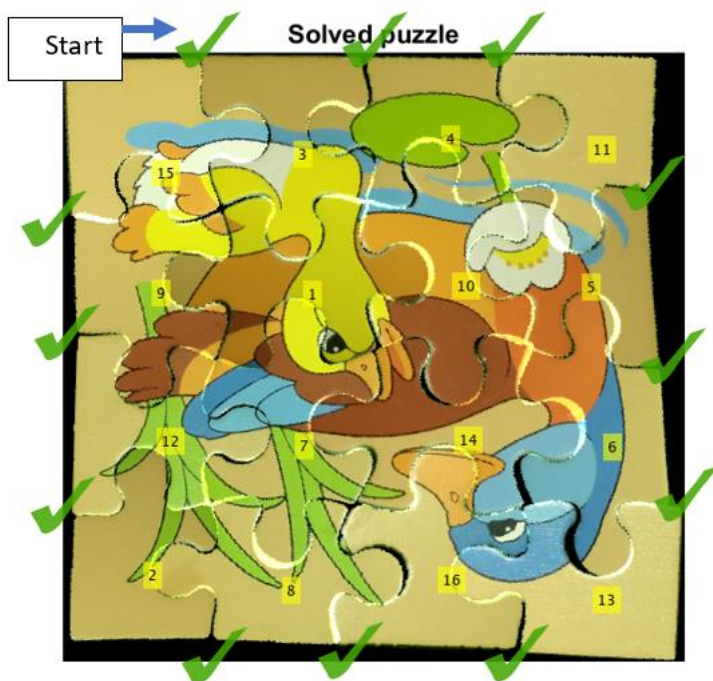
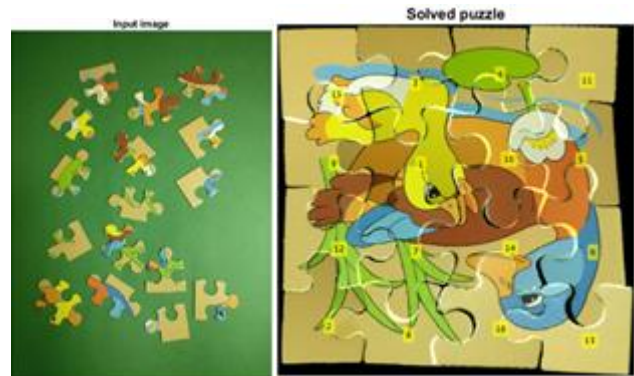
בנוסף, החלק המתאים בהיר יותר לעומת החלק מספר 1 ולכן המחיר שלו גבוה מבין האפשרויות לחיבור.

הצגת החלקים האפשריים לחיבור במיקום זה והמחירים עבור כל חלק:

Piece 1 with 16	Piece 1 with 8	Piece 1 with 6
		
-0.1436	-0.1608	-0.2402

נציין, כי החלק המתאים לפתרון מקבל את המחיר הגבוה ביותר מבין שלושת החלקים, ולכן בבניית העץ אנו אפילו לא מתייחסים אליו בחישוב הפתרון הכולל. לכן אין שום אפשרות של תיקון עתידי על ידי שאר חיבורי המסגרת.

מקרה 3 - פתרון מושלם:



במקרה זה קיבלנו 100% הצלחה. כפי שכבר ציינו, גם כאן, כאשר האלגוריתם פתר נכון את המסגרת אז גם החלקים הפנימיים נבחרים בצורה טובה.

מחשוב כולל של ממוצעי ההצלחות בדוגמאות שהצגנו, הצלחנו להגיע לאחוזי דיוק של כ-80%.

מסקנות

- ניתן להסיק מהצלחת האלגוריתם כי שימוש בפתרון מבוסס צורה ופתרון מבוסס צבע הכרחיים שכן שילוב ביניהם נותן לנו את הפתרון הטוב ביותר בתנאי המערכת הנתונים.
- באלגוריתם יש trade off בין מהירות החישוב לבין אחוזי הדיוק. היות והגבלנו את עץ הפתרונות להיות בינארי והגבלנו את מספר הבחירות של החלק השני הכי טוב קיבלנו מהירות חישוב סבירה יחד עם אחוזי דיוק טובים. כשניסינו להגדיל את העץ קיבלנו אחוזי דיוק גבוהים יותר אך זמן החישוב עלה משמעותית.
- התאורה היא גורם מרכזי לטעויות בתוצאה. כאשר יש הבדלי תאורה גדולים בין החלקים, או שיש צל של החלקים על הרקע אז מתקבלות תוצאות לא טובות או אפילו בעיות סגמנטציה.

סיכום

הפרויקט שבחרנו לעצמנו טמן אתגרים תכנותיים רבים אשר בעזרת הכלים שלמדנו וכלים שפיתחנו נהיו ממשיים ואפשריים. הפרויקט לא היה פשוט והצריך מאיתנו זמן, מאמץ ולמידה רחבה ובאמצעות עבודה משותפת הצלחנו להשיג את מטרותינו המקורית אשר הייתה לפתח פרויקט המאפשר פתרון אוטומטי לפאזלים.

ראוי לציין, כי במהלך התהליך רעיונות רבים הועלו אך בשל היקף הפרויקט היינו צריכים למקד את עצמנו לכיוון המטרה המשותפת ולא להתפזר לכיוונים אחרים, שכן רעיונות אלו יכלו לשפר את האלגוריתם אך היו מצריכים זמן פיתוח ובדיקות נרחבים.

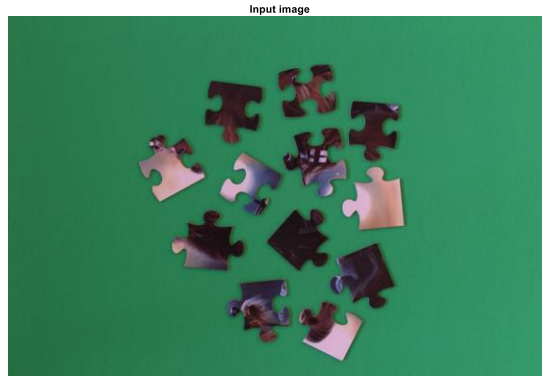
בסך הכל, אנחנו מרוצים מהתוצאה שהתקבלה שכן בעלת אחוזי הצלחה ודיוק טובים ואיפשרו לנו להיות גאים בתוצר הסופי שלנו שכן הצליח להתמודד עם יותר מפאזל אחד.

References

- [1] Automatic Jigsaw Puzzle Solver -Nithyanada Bhat Kumbala,
https://nithyanandabhat.weebly.com/uploads/4/5/6/1/45617813/project_report-jigsaw-puzzle.pdf
- [2] Initial flow ideas- <https://www.abtosoftware.com/blog/computer-vision-powers-automatic-jigsaw-puzzle-solver>
- [3] https://web.stanford.edu/class/cs231a/prev_projects_2016/computer-vision-solve_1.pdf

נספח א'- פתרון פאזל נוסף – צ'ובקה

נציג פאזל נוסף שאיתו האלגוריתם מתמודד בצורה טובה ומציג פתרון מושלם של הפאזל.
נציין זה כי בפאזל זה יש פחות חלקים ולכן זמן הריצה קטן ומספר האפשרויות לטעות קטן.



כפי שציינו בשלב הסגמנטציה, יצרנו מודל GMM נוסף עבור פאזל זה.

וזו התוצאה שהתקבלה:

