**ספירת ספרות :**

הפרויקט שלנו מציע מערכת חכמה כדי לעזור לפתור חידות. מטרת הפרויקט שלנו היא לתמוך במשתמש בצורה מתודולוגית ואינטואיטיבית בפתרון הפאזל, מבלי שהמערכת תפתור את הפאזל עצמו. כתוצאה מכך, הפרויקט שלנו יהיה חינוכי מטבעו ויכול להועיל לאנשים שרוצים לעורר את כישורי פתרון הבעיות שלהם באמצעות למידה מתודולוגית לילדים ומבוגרים כאחד.  
כשעברנו על העיצוב של הפרויקט שלנו, הגענו למסקנה שכדי לעזור למשתמש בצורה מתאימה, נשתמש בעצם בשתי פונקציות שמובילות את המשתמש לפתור את הפאזל בעצמו. ראשית אנו מעודדים את המשתמש להתחיל על ידי יצירת קווי המתאר של הפאזל. במלים אחרות אנו מלווים בחלק הראשון את המשתמש למיין את חלקי הפאזל על ידי הפרדת החלקים הפנימיים והחלקים הממוקמים על מתאר הפאזל. כדי לעשות זאת, המשתמש יצטרך לשלוח את חלקי הפאזל למערכת בתמונות. משם, הגיעו מספר שאלות, כיצד תוכל המערכת לזהות את החלקים המוצגים בתמונה? ולבסוף, באיזה אלגוריתם עלינו להשתמש כדי למיין את החלקים לפי צורתם? כדי ליישם את הפונקציונליות הזו, חיוני שתהיה לך ספריית ראיית מחשב שיכולה לנתח ולעבד תמונות בצורה מדויקת ומהירה. לאחר ניתוח מעמיק, בחרנו ב- OpenCV, ספרייה חזקה ורב-תכליתית המשמשת ביישומי עיבוד תמונות וראיית מחשבים.

מאמר רלוונטי במיוחד לפרויקט שלנו [1] ממחיש את השימוש ב- OpenCV לניתוח חלקי פאזל. מדריך זה מדגיש את השלבים הדרושים כדי לבודד חלקים של תמונה, לזהות את קווי המתאר שלהם ולסווג אותם על פי המאפיינים הגיאומטריים שלהם. דוגמה קונקרטית זו מדגימה כיצד ניתן ליישם את OpenCV לביצוע משימות דומות למטרה הראשונה שלנו: הפרדת החלקים הפנימיים מאלה הממוקמים על קווי המתאר של הפאזל.  
המאמר מציג טכניקות כגון זיהוי מיתאר באמצעות הפונקציה CV2.FindContours, סינון חלקים המבוססים על תכונות גאומטריות (כגון נוכחות של קצוות ישרים עבור חלקים של קווי המתאר), ויישום מסכות כדי לבודד חלקים רלוונטיים. תכונות אלה הופכות את OpenCV לבחירה מתאימה ליישום המערכת שלנו, ומבטיחות דיוק ויעילות בעיבוד תמונות המסופקות על ידי המשתמש.

בשלב השני המערכת שלנו תצטרך למיין את החלקים הנותרים של הפאזל על פי הצבעים שלהם כדי להקל על פתרון הפאזל. לשם כך, חשבנו תחילה על יצירת אלגוריתם שמזהה את הצבע של כל יצירה ומציב אותה בקבוצות שונות. הגענו למסקנה כי עבור חלקים מרובי צבעים יהיה קשה להבחין ביניהם על פי הצבעים שלהם. היה עלינו לחשוב על חלופה אחרת למיון החלקים לפי צבעיהם. לאחר מחקר מסוים היה לנו הרעיון לנתח את התמונה המקורית של הפאזל, לעשות עיבוד תמונה על תמונה זו כדי להבחין בין האובייקטים השונים הנוכחים בתמונה, על ידי אלגוריתם של פילוח לפי תמונה [2]. כל אובייקט מייצג חלק חיוני בפאזל (אדם, שמיים, ים, עץ...) [2, איור 1]. כל אובייקט נתפס יוצר קבוצת צבעים. המערכת תנתח את התמונה של חלקי הפאזל ותשווה כל חתיכה עם כל אובייקט מזוהה בתמונה. על מנת לבצע השוואות אלה נבקש תיאורים כמו SIFTכפי שהוא מתאים ביותר להשוואת תמונה תחת שינוי קנה מידה וסיבוב [3].  
אם חלקx מתאימה לאובייקט y, אותן חלק תוקצה לקבוצת הצבעים השייכת לאותו אובייקט ,y וכן הלאה עד לניתוח של כל החלקים.

עבור השלב האחרון המוצע על ידי המערכת שלנו, עכשיו כאשר קבוצות הצבעים נוצרים, אנו מציעים לסווג בכל קבוצת צבע את החלקים לפי צורתם (אם הם מעוקלים פעם, פעמיים, שלוש פעמים או ארבע פעמים). לשם כך נשתמש באותה שיטת רזולוציה שבה נעשה שימוש בשלב הראשון, כאשר היה צורך ליצור את קווי המתאר של הפאזל על ידי הפרדת החלקים הפנימיים מהפאזל והאחד הממוקם על המתאר.

**ביבליוגרפיה:**

**[1] :Solving Jigsaw Puzzles with Python and OpenCV"**

[Solving Jigsaw puzzles with Python and OpenCV | by Riccardo Albertazzi | Towards Data Science](https://towardsdatascience.com/solving-jigsaw-puzzles-with-python-and-opencv-d775ba730660)

**[2] : Insights into Image Understanding: Segmentation Methods for Object Recognition and Scene Classification**

<https://www.mdpi.com/1999-4893/17/5/189>

**[3]: A Comparative Study of SIFT and its Variants**

[msr-2013-0021](https://intapi.sciendo.com/pdf/10.2478/msr-2013-0021)