

9. Computer Vision

9.2 Segmentation

אחד האתגרים הכי משמעותיים בעולם הראייה הממוחשבת הוא זיהוי אובייקטים בתמונה והבנת המתרחש בה. אחת הטכניקות הקלאסיות להתמודדות עם משימה זו הינה ביצוע סגמנטציה, כלומר, התאמת label לכל פיקסל בתמונה. בתהליך הסגמנטציה מבצעים חלוקה/בידול בין עצמים שונים בתמונה המצולמת באמצעות סיווג ברמת הפיקסל, כלומר כל פיקסל בתמונה יסווג וישויך למחלקה מסוימת.

ישנם שימושים מגוונים באלגוריתמים של סגמנטציה – הפרדה של עצמים מסוימים מהרקע שמאחוריהם, מציאת קשרים בין עצמים ועוד. לדוגמה, תוכנות של שיחות ועידה, כמו zoom, skype, teams וכדו', מאפשרות בחירת רקעים שונים עבור המשתמש, כאשר מלבד הרקע הנבחר רק הגוף של המשתתף מוצג בווידאו. הפרדת גוף האדם מהרקע והטמעת רקע אחר מתבצעות באמצעות אלגוריתמים של סגמנטציה. דוגמה נוספת – ניתן לזהות בתמונה אדם, כלב, וביניהם רצועה, ומכך ניתן להסיק שתוכן התמונה הוא אדם מחזיק כלב בעזרת רצועה. במקרה זה, הסגמנטציה נועדה למצוא קשר בין עצמים ולהבין את המתרחש.

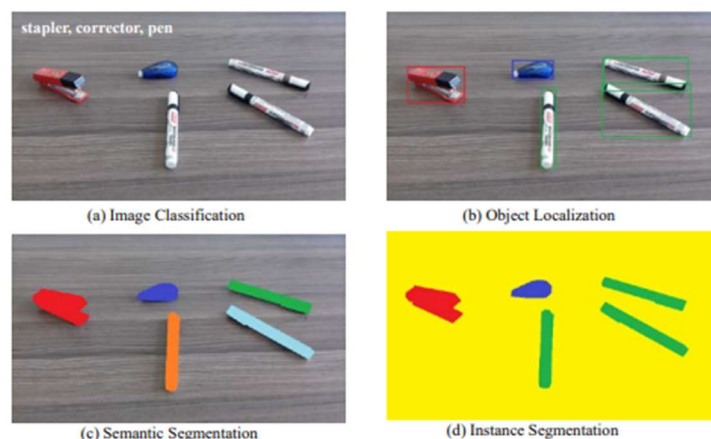
9.2.1 Semantic Segmentation vs. Instance Segmentation

קיימים שני סוגים עיקריים של סגמנטציה:

Semantic segmentation (חלוקה סמנטית) - חלוקה של כל פיקסל בתמונה למחלקה אליה העצם אותו הוא מייצג שייך. למשל, פיקסל יכול להיות משויך לכלי רכב, בן אנוש, מבנה וכדו'.

Instance segmentation (חלוקה מופעית) - חלוקה של פיקסל בתמונה למופע של אותה מחלקה אליה העצם אותו הוא מייצג שייך. במקרה זה, בתמונה בה מופיעים מספר כלי רכב, תבוצע חלוקה של כל פיקסל לאיזה כלי רכב אותו פיקסל מייצג – מכונית 1, מכונית 2, אופנוע 1, משאית 1 וכדו'.

ההבדל העיקרי בין שני סוגים אלו הוא ברמת עומק המיפוי של פיקסל - המיפוי עשוי לסווג את הפיקסל למחלקה כלשהי, או לעצם ספציפי בתמונה. עומק המיפוי משליך גם על עלות המיפוי. החלוקה הסמנטית מבוצעת ישירות, בעוד שהחלוקה המופעית דורשת בנוסף ביצוע של זיהוי אובייקטים כדי לסווג מופעים שונים של המחלקות.



איור 9.2.1 משימות שונות תחת התחום של Computer Vision. ניתן להבחין בהבדל שבין Semantic segmentation (התאמת כל פיקסל למחלקה מסוימת) לבין Instance segmentation (התאמת כל פיקסל למופע של מחלקה מסוימת).

9.3 Face Recognition and Pose Estimation

9.3.1 Face Recognition

אחד מהיישומים החשובים בעיבוד תמונה הינו זיהוי פנים, כאשר ניתן לחלק משימה זו לשלושה שלבים:

1. Detection – מציאת הפרצופים בתמונה.
2. Embedding – מיפוי כל פרצוף למרחב חדש, בו הפיצ'רים שאינם קשורים לתיאור הפנים (למשל: זווית, מיקום, תאורה וכדו') אינם משפיעים על הייצוג.
3. Searching – חיפוש במאגר של תמונות למציאת תמונות פנים הקרובה לתמונת הפנים שחולצה מהתמונה המקורית.

גישה פשטנית, כמו למשל בניית מסווג המכיל מספר יציאות כמספר הפנים אותם רוצים לזהות, הינה בעייתית משתי סיבות עיקריות: ראשית יש צורך באלפי דוגמאות לכל אדם (שלא ניתן בהכרח להשיג). כמו כן, נצטרך ללמד את המערכת מחדש בכל פעם שרוצים להוסיף משימה חדש. כדי להתגבר על בעיות אלו מבצעים "למידת מטריקה" (metric learning) בה מזקקים פיצ'רים של פנים ויוצרים וקטור יחסית קצר, למשל באורך 128, המכיל את האלמנטים המרכזיים בתמונות הפנים. כעת נפרט את שלושת השלבים:

1. מציאת פנים.

כדי למצוא פרצופים בתמונה ניתן להשתמש ברשתות המבצעות detection, כפי שתואר בפרק 9.1. שיטה מקובלת למשימה זו הינה Yolo, המבוססת על חלוקת התמונה למשבצות, כאשר עבור כל משבצת בוחנים האם יש בה אובייקט מסוים, מהו אותו אובייקט, ומה ה-bounding box שלו.

2. תיאור פנים.

כאמור, המשימה בתיאור פנים נעשית בעזרת metric learning, כאשר הרעיון הוא לזקק פנים לוקטור שאינו מושפע מפיצ'רים שלא שייכים באופן מהותי לפנים הספציפיות האלה, כגון זווית צילום, רמת תאורה וכדו'. בכדי לעשות זאת יש לבנות רשת המקבלת פנים של בנאדם ומחזירה וקטור, כאשר הדרישה היא שעבור שתי תמונות של אותו אדם יתקבלו וקטורים מאוד דומים, ועבור פרצופים של אנשים שונים יתקבלו וקטורים שונים. למעשה, פונקציית ה-loss תקבל בכל פעם minibatch, ותעניש בהתאם לקרבה בין וקטורים של אנשים שונים וריחוק בין וקטורים של אותו אדם.

כעת נניח שיש לנו קלט X , המכיל אוסף פרצופים. כל איש יסומן באות אחרת – A, B, C, ותמונות שונות של אותו אדם יסומנו על ידי אות ומספר, כך שלמשל X_{A1} זוהי התמונה הראשונה של אדם A בסט הקלט X , וכמובן ש- X_{A1} ו- X_{A2} הן שתי תמונות של אותו אדם. באופן גרפי, בדו-ממד ניתן לתאר זאת כך (בפועל הווקטורים המייצגים פנים יהיו בממד גבוה יותר):



איור 9.1 (a) דוגמאות מסט הפרצופים X . (b) איך נרצה שהדאטה ימופה לממד חדש Y .

כאמור, נרצה לבנות פונקציית loss שמעודדת קרבה בין X_{A1} ו- X_{A2} , וריחוק בין X_{A1} ו- X_{B1} . פונקציית ה-loss מורכבת משני איברים, המודדים מרחק אוקלידי בין וקטורים שונים:

$$L = \sum_X \|Y(X^{Ai}) - Y(X^{Aj})\| - \|Y(X^{Ai}) - Y(X^{Bj})\|$$

כאשר האיבר הראשון ינסה להביא למינימום וקטורים של אותו אדם, והאיבר השני ינסה להביא למקסימום וקטורים של פרצופים שאינם שייכים לאותו אדם. כיוון שנרצה להימנע מקבלת ערכים שליליים, נוסיף פונקציית מקסימום. בנוסף, ניתן 'להרחיק' תוצאות של פרצופים שונים על ידי הוספת קבוע k , כך שהפרש בין המרחק של פרצופים של אנשים שונים לבין המרחק של פרצופים של אותו איש יהיה לפחות k :

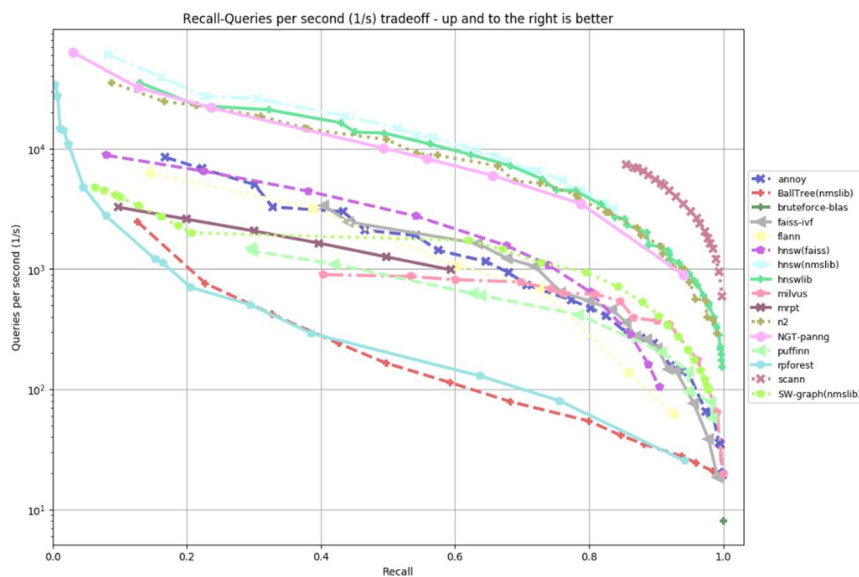
$$L = \sum_X \max(\|Y(X^{Ai}) - Y(X^{Aj})\| - \|Y(X^{Ai}) - Y(X^{Bj})\| + k, 0)$$

loss כזה נקרא triplet loss, כיוון שיש לו שלושה איברי קלט – שתי תמונות של אותו אדם ואחת של משימה אחר. כאמור, הפלט של הרשת הנלמדת צריך להיות וקטור המאפיין פנים של אדם, ומטרת הרשת היא למפות פרצופים

שונים של אותו אדם לוקטורים דומים עד כמה שניתן, ואילו פרצופים של אנשים שונים יקבלו וקטורים רחוקים זה מזה.

3. מציאת האדם

בשלב הקודם, בו התבצע האימון, יצרנו למעשה מאגר של פרצופים במרחב חדש. כעת כשיגיע פרצוף חדש, כל שנותר זה למפות אותו למרחב החדש, ולחפש במרחב זה את הווקטור הקרוב ביותר ולקטור המייצג את הפנים החדשות. בכדי לעשות זאת ניתן להשתמש בשיטות קלאסיות של machine learning, כמו למשל חיפוש שכן קרוב (כפי שהוסבר בחלק 2.1.3). שיטות אלו יכולות להיות איטיות עבור מאגרים המכילים מיליוני וקטורים, וישנן שיטות חיפוש מהירות יותר (ובדרך כלל המהירות באה על חשבון הדיוק). בעזרת השיטה המובילה כרגע (SCANN) ניתן להגיע לכמה מאות חיפושים שלמים בשנייה (החיפוש ב-100 ממדים מתוך מאגר של 10000 דוגמות).



איור 9.2 עבור פרצוף נתון, מחפשים עבורו וקטור תואם בממד החדש המכיל ייצוג וקטורי של הפרצופים הידועים. בכל שיטה יש טרייד-אוף בין מהירות החיפוש לבין הדיוק, ובגרף זה מוצגות שיטות שונות.

מלבד זיהוי וסיווג פנים, יש גם שיטות של מציאת אלמנטים של פנים הכוללות אף, עיניים וכו'. אחת השיטות המקובלות משתמשת בשערוך הצורה של פנים אנושיות, וניסיון למצוא את איברי הפנים לפי הצורה הסטנדרטית. בשיטה זו ראשית מבצעים יישור של הפנים והתאמה לסקאלה אנושית (על פי מרחק בין האיברים השונים בפנים), ולאחר מכן מטילים 68 נקודות מרכזיות על התמונה המיושרת, מתוך ניסיון להתאים בין הצורה הידועה לבין התמונה המבוקשת.



איור 9.3 זיהוי אזורי פנים של אדם על ידי התאמת פנים לסקאלה אנושית והשוואה למבנה של פנים המכיל 68 נקודות מרכזיות.

9. References

Segmentation:

<https://arxiv.org/ftp/arxiv/papers/2007/2007.00047.pdf>

Face recognition:

https://docs.opencv.org/master/d2/d42/tutorial_face_landmark_detection_in_an_image.html

<http://blog.dlib.net/2014/08/real-time-face-pose-estimation.html>