

אפליקציה לזיהוי פנים

Face Finder

שם המגיש: יבגני גולדשטיין

תיאור הפרויקט:

אפליקציה לזיהוי פנים FACE FINDER. אפליקציה שמומשה כולה בשפת תכנות JAVA, תוך כדי דגש על ביצוע חישובים יעילים, ואינטרפייס ידידותי ושימושי למשתמש.

רעיונות ומטרות:

נושא של זיהוי פנים ממוחשב סיקרן אותי מאז שראיתי את זה בסרט בתור ילד. ואחרי שלמדתי בקורס ונחשפתי לאתגרים הרבים שצריך להתמודד איתם על מנת לעשות זאת, הופתעתי לגלות שהשיטה יחסית לא מסובכת, הרגשתי שאני הולך להנות מתהליך פיתוח של אפליקציה כזאת, כמו כן אוכל לספק סקרנות שלי לגבי שאלות אחרות.

תוך כדי הקורס כתבנו תכניות שממשות אלגוריתמים מתחום ריאה חישובית בסביבת פיתוח MATLAB, כאשר חוזרת טענה שזהו כלי מאוד חזק ונוח לבצע חישובים מתמטיים מורכבים. במהלך הקורס לא יכלתי להמנע מלהשוות את MATLAB לסביבות פיתוח שאני רגיל לעבוד איתם ולהשוות מבחינת יכולות וביצועים. הרושם שנוצר לי תוך כדי הקורס היה הפוך ממה שטוענים, כאשר צריך להדגיש שלא יצא לי לפתח קוד משמעותי שדורש חישובים מתמטיים רבים בסביבת פיתוח אחרות. לכן שמחתי לנצל הזדמנות זו.

האפליקציה מומשה בשפת תכנות JAVA בסביבת פיתוח INTELLIJE, כאשר שמתי דגש על הנדסת תוכנה הכי טובה שיכלתי לעשות, תוך כדי שימוש בסטנדרטים וכלים השימושיים היום בתחום שאני מכיר ולחן זמנים מאפשר לי ללמוד (כמו למשל שימוש ב XML, SWING, JAR, OPENCV...).

מעבר לפיתוח אפליקציה שמתי למטרה להתרשם מהכלים ש JAVA נותנת למפתח אפליקציה מסוג זה, עד כמה קל ללמוד את הכלים ולמצוא פתרונות לבעיות הנוצרות (תיעוד, תמיכה ברשת), עד כמה מהר אני יכול לפתח אפליקציה מסוג זה, יעילות של קוד (מבחינת זמן ריצה ניצול משאבים...).

מימוש:

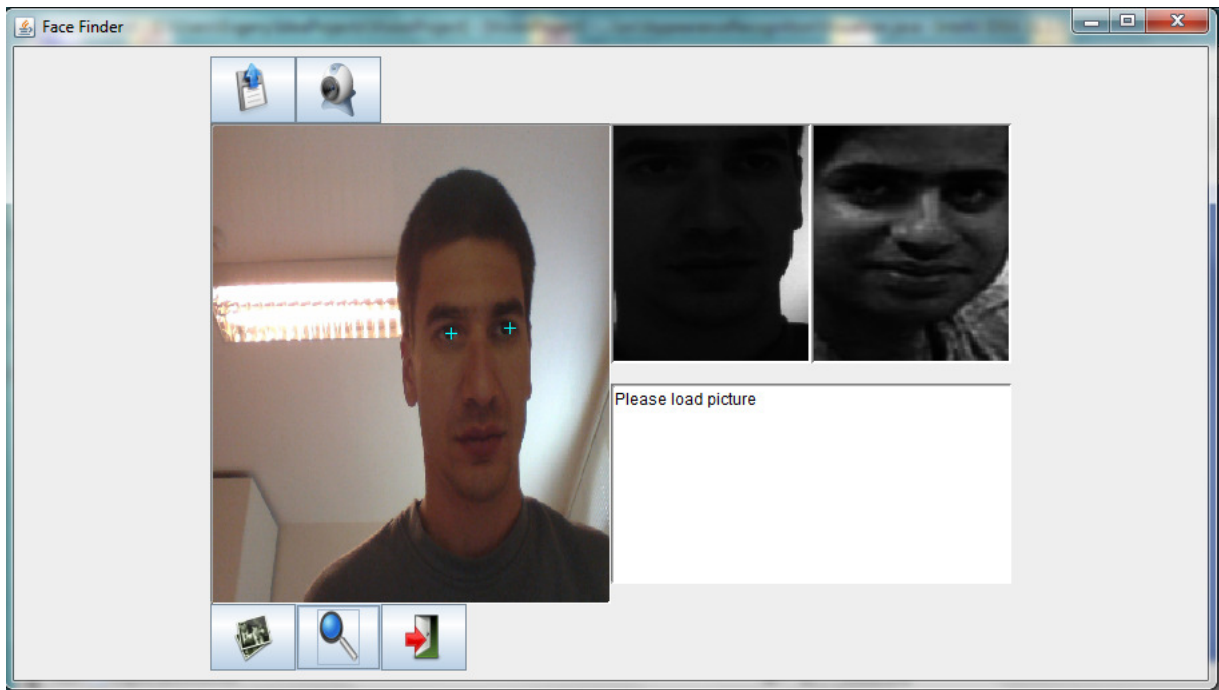
המימוש הוא של אלגוריתם שמתואר בפירוט במאמרים הבאים:

<http://www.scholarpedia.org/article/Eigenfaces>

<http://en.wikipedia.org/wiki/Eigenface>

האפליקציה מגיע בצורה של סיפריה מקובצת שמכילה קובץ JAR עם קוד JAVA וספריית תמונות, קובץ קונפיגורציה (config.xml) שמאפשר למשתמש להגדיר את סיפריית התמונות שאיתה הוא רוצה לעבוד, כמות הווקטורים העצמיים של המערכת....) וסקריפט הרצה של האפליקציה.

תוצאות:



תאור קצר של האפליקציה

לאפליקציה יש בסיס נתונים של תמונות כאשר האפליקציה עולה, על מנת לחפש התאמה בבסיס נתונים, צריך לטעון לאפליקציה תמונה ברזולוציה מתאימה (רזולוציה ניתן לשנות בקובץ קונפיגורציה), ואת זה אפשר לעשות על ידי טעינת תמונה כלשהיא לאפליקציה (כפתור ימני מלמעלה) או על ידי צילום תמונה ממצלמה של המחשב. לאחר טעינת תמונה לחלון המרכזי צריך לבחור פנים שרוצים לחפש, על ידי סימון עיניים על התמונה (כפתור ימני על העכבר) ולאחר מכן לחיצה על כפתור השמאלי של העכבר על מנת להשלים את הגזירה של הפנים בצורה מתאימה. כאשר התמונה בחלון האמצעי נראת לנו סבירה (אפשר לחזור על תהליך הגזירה כמה פעמים) אנו יכולים להוסיף אותה לבסיס נתונים או לחפש התאמה בבסיס נתונים על ידי לחיצה על כפתורים ימני או אמצעי בתחתית המסך.

מסקנות:

- מבחינת התמודדות עם האלגוריתם, יש המון חומר ברשת שמסביר בצורה נוחה את האלגוריתם, אבל על מנת להבין לרמה שאוכל לממש לקח כמה ניסיונות לא מוצלחים.

- כאשר הגעתי לשלב עיצוב של האפלקציה, היה ברור לי שצוואר בקבוק (מבחינת ביצועים, וזמן פיתוח) יהיה בחלקים המתמטיים שדורשים עבודה עם מטריצות. אחד ההבדלים בין JAVA ל MATLAB הוא בזה שקיימים הרבה פתרונות דומים (אני מצאתי בקלות 5 סיפריות שונות לעבודה עם מטריצות) של אירגונים שונים. על מנת לבחור את הסיפריה האופטימלית מבחינת המטרות שהצבתי היה צריך להכיר טוב את יתרונות והחיסרונות של כל אחד מהם (מה שדורש ניסיון בעבודה עם כלים כאלה, או ביצוע חקירה מעמיקה על כל אחד מהם).
- בניגוד לעבודה עם MATLAB נתקלתי בכמות מוגזמת של תיעוד עד רמה שהיה לי קשה למצוא פתרונות לבעיות שנתקלתי בהם.
- הרבה פעמים היו חסרים לי את הכלים של MATLAB שמאפשרים לבדוק את תקינות החישובים (הגרידים למיניהם).
- בין 15 ל 20 אחוז מהפעולות על המריצות שהייתי צריך לעשות הייתי צריך לממש בעצמי, חלק מהדברים בוודאי קיים ב MATLAB.
- מאוד השתדלתי לתת אפשרות לעבוד עם מצלמה באפלקציה, למרות שקיימים הרבה פתרונות טובים, אחרי שניסיתי 4 סיפריות שונות, בקושי הצלחתי להפעיל אחד מהם, מה שמאוד הפתיע אותי.

המסקנה העיקרית היא, שעל מנת לפתח אפלקציה מהסוג הזה ברמה מקצועית צריך הרבה ניסיון, (על מנת להכיר לפחות חלק מהכלים הקיימים ב JAVA), אבל אם כן מכירים את התחום אפשר להנות מיתרונות רבים של פיתוח ב JAVA, וכמובן שלמרות שאני חושב שיש מה להוסיף ל MATLAB אני רכשתי הערכה רבה למערכת זו.

מקורות:

<http://www.scholarpedia.org/article/Eigenfaces>
<http://en.wikipedia.org/wiki/Eigenface>
<https://acs.lbl.gov/software/colt/api/cern/colt/matrix/package-summary.html>
<https://code.google.com/p/javacv/>