

השוואת פריסות מקלדת באמצעות ראייה ממוחשבת

מבוא

פרויקט זה מיישם מערכת אוטומטית להתאמת מקשים בין פריסות מקלדת שונות באמצעות טכניקות ראייה ממוחשבת. המערכת מסוגלת להשוות בין מקלדות Mac, ו Windows-וכן בין וריאציות שונות של מקלדות, למשל, מקלדת לבנה (שבה האותיות העבריות ממוקמות למטה) לעומת מקלדת ורודה (שבה סדר האותיות הפוך: האותיות העבריות ממוקמות מעל האותיות באנגלית).

שיטות וטכניקות

1. עיבוד מקדים של התמונה

המערכת מבצעת מספר שלבי עיבוד מקדים על תמונות המקלדת:

- **המרה בינארית:** התמונות מומרות לפורמט בינארי באמצעות סף אדפטיבי :
 - מקלדת Mac: ערך סף 135
 - מקלדת Windows: ערך סף 170
 - מקלדת Mac ורודה: ערך סף 100
- **פעולות מורפולוגיות:** מיושמות פעולות סגירה (closing) להפחתת רעש וטיוב התמונה :
 - שימוש בקרנל בגודל 3×3
 - ביצוע פעולת הסגירה פעמיים לקבלת תוצאות מיטביות

2. זיהוי וניתוח של מקשי המקלדות באמצעות קונטורים

המערכת מזהה ומנתחת את חלקי המקלדת באמצעות:

- **זיהוי קונטורים:** שימוש באלגוריתם `findContours` של OpenCV
 - סינון קונטורים קטנים (שטח מינימלי: 100 פיקסלים)
 - הוספת ריפוד של 2 פיקסלים מסביב לכל חלק

3. זיהוי קונטורים המוכלים בתוך כל מקש

- מניעת כפילויות באמצעות בדיקת הכלה
- מיון לפי גודל השטח בסדר יורד (מהשטח הגדול לקטן)

- בדיקת נקודת מרכז, אורך ורוחב עבור כל קונטור

4. התאמת מקשים

ההתאמה מתבצעת באמצעות פונקציית **Feature Matching** המהווה את ליבת המערכת:

- שימוש באלגוריתם: **SIFT**

- זיהוי נקודות עניין תלויות בסקאלה וברוטציה
- חילוץ וקטורי תיאור (descriptors) עבור כל נקודת עניין

- סינון התאמות:

- סינון באמצעות ערך סף של 0.875
- התאמת קונטורים ספציפית (כל קונטור במקלדת המקור יכול להתאים לקונטור יחיד במקלדת היעד)

5. בחירת התאמות חכמה

המערכת מיישמת מנגנון מתוחכם לבחירת ההתאמות הטובות ביותר:

- מערכת ניקוד מבוססת פערים:

- פער מקסימלי: 2.5 להשוואת Mac-Windows
- פער מקסימלי: 1.0 להשוואת Mac-Mac
- שמירת עד 3 ניקודים (Scores) שונים בטווח הפער
- הגבלה ל-5 התאמות מקסימליות לכל חלק

- סינון התאמות כפולות:

- מעקב אחר אינדקסים של חלקי Windows שכבר נמצאו
- אימות ציון סף מינימלי (4.5)
- שמירה על התאמות ייחודיות

תצוגה

לאחר ביצוע ההתאמות, המערכת מציגה את התוצאות עבור כל מקש במקלדת המקור. התהליך מתבצע באמצעות הפונקציה- `visualize_matches_detailed`, אשר יוצרת תצוגה מפורטת של ההתאמות:

- שורה ראשונה: שורת סימון מיקומים במקלדת המקור: סימון מיקום ההתאמה במקלדת המקור באמצעות קונטורים וציור מסגרת ירוקה מסביב לאזור שהתאים.
- שורה שנייה: שורת התאמות תכונות: הצגת התאמות בין מקשי המקלדת המקוריים והיעדיים תוך שימוש באלגוריתם `drawMatches` (מקש המקור תמיד יימצא בצד שמאל ועל ידו בצד ימין המקש שנמצא מתאים לו ממקלדת היעד) תוך הצגת ציון ההתאמה לכל התאמה.
- שורה שנייה: שורת סימון מיקומים במקלדת היעד: סימון מיקום ההתאמה במקלדת היעד באמצעות קונטורים וציור מסגרת ירוקה מסביב לאזור שהתאים.
- סידור גרפי אוטומטי: התאמות מרובות מוצגות במערך של תמונות, כך שהתוצאה נוחה להבנה ומסודרת.

◆ בנוסף, מצורפים שני קבצי gif שמציגים את התוצאות:

1. `keyboard_matches_pinc_vs_white.gif`: מציג את התוצאות של השוואת מקלדת mac ורודה (המקור) שבה המקשים מסודרים כך שהאותיות באנגלית למטה, והאותיות עברית למעלה למקלדת mac לבנה (היעד) שבה המקשים מסודרים הפוך (עברית למטה ואנגלית למעלה).
2. `keyboard_matches_win_vs_mac.gif`: מציג את התוצאות של השוואת מקלדת mac למקלדת windows.

סיכום

המערכת מציגה פתרון מקיף להשוואת פריסות מקלדת באמצעות שילוב של טכניקות ראייה ממוחשבת מתקדמות. היא מספקת תוצאות מדויקות ואמינות, תוך התמודדות עם אתגרים כמו שינויי תאורה, הבדלי זווית ווריאציות בעיצוב המקלדת.

הגישה המודולרית והשימוש באלגוריתמים מתקדמים כמו SIFT מאפשרים גמישות והתאמה למגוון רחב של סוגי מקלדות, תוך שמירה על דיוק ואמינות בתהליך ההתאמה.

- בשביל להריץ את הקוד שמוצא התאמות בין שני המקשים יש לעשות את השינויים הבאים בחלי-
ה- `main` `:(if __name__ == "__main__")`:
1. יש להוריד מהערה את שורה 498 בקוד:

```
# main(pink_mac_binary, mac_binary, mac, pink_mac,max_gap_thresh=1.0)
```

2. ולשים בהערה את שורה 480 בקוד:

```
main(mac_binary, windows_binary, windows, mac)
```

