דוח עבור מטלת DS – נועם דיאמנט

הקדמה

לפני שנצלול לכל שלב בתהליך שביצעתי, אציג את כל השלבים בקצרה.

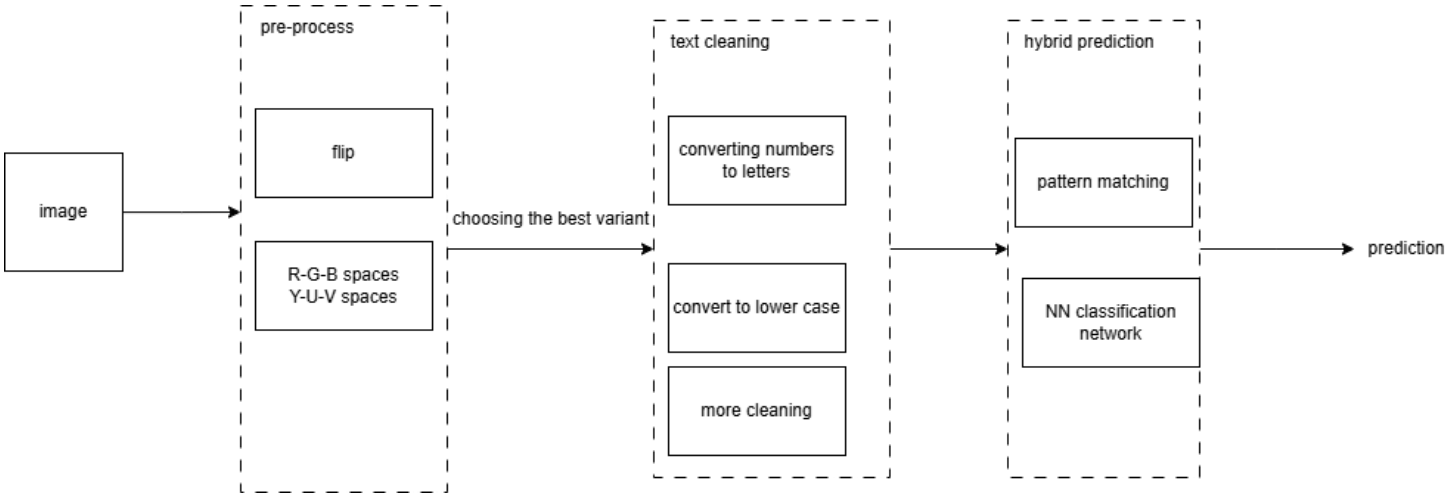
בתחילה, ביצעתי חקירה של הדאטה שקיבלתי, בחנתי את הדאטה, את המבנה שלו, והבחנתי בבעיות שאצטרך לפתור. לאחר שעברתי על בפעם הראשונה על הדאטה, התחלתי לנסות ולנתח אותו בצורות שונות כדי לחלץ ממנו את הטקסט בצורה הטובה ביותר. זה היה תהליך שבמהלכו ניסיתי כמה וכמה פתרונות עד שהגעתי לפתרון הטוב ביותר שהגעתי אליו.

לאחר שסיימתי לעבוד על הדאטה בתור תמונות, עברתי לעבוד על הטקסט שהתקבל מהתמונות. ביצעתי כמה צעדים של ניקוי וסידור של הטקסטים שהתקבלו, כדי שהמסווג שאפעיל יסווג את הדאטה בצורה הטובה ביותר.

לבסוף, בחנתי שני סוגים מסווגים ובחרתי את הטוב מביניהם.

אציין שרוב הזמן שהשקעתי במטלה הושקע בחקירה ובניתוח של התמונות כדי להוציא מהם את הטקסטים בצורה הטובה ביותר.

אציג את התהליך הכללי הסופי של הפתרון שלי (נחזור אליו בסוף, אבל במהלך הדו"ח אני מציג את הדרכים שהובילו אותי לפתרון הזה):



חקירת הדאטה

בתור שלב ראשון, הסתכלתי על הדאטה. כתבתי סקריפט במחברת שמאפשר לי לראות כל פעם כמה תמונות אקראיות מכל תיקייה, והתבוננתי בתמונות שהתקבלו. הסתכלתי גם בעצמי בתוך התיקיות על התמונות כדי לקבל מבט נוסף על התמונות. שמתי לב לכמה דברים שאצטרך להתמודד איתם:

תמונות הפוכות, תמונות שהכיתוב בהם דומה לרקע או שהכיתוב לא ברור – עם הדברים הללו התמודדתי ברמת התמונה, קרי ברמת העיבוד שביצעתי על התמונות.

טקסט מפוצל (מילה אחת שמפוצלת לשתיים עם רווח באמצע), אותיות גדולות וקטנות בטקסט, מספרים בטקסט במקום אותיות, טקסט לא קשור – עם הדברים הללו התמודדתי ברמת הטקסט, קרי ברמת הניקוי והסידור של הטקסט לאחר שחילצתי אותו מהתמונות.

התמודדות ברמת התמונות:

1. תמונות הפוכות
2. תמונות עם כיתוב דומה לרקע, כיתוב לא מספיק בולט

דבר ראשון ששמתי אליו, הוא שיש הרבה תמונות הפוכות, כלומר תמונות שהכיתוב בהן הפוך. הפתרון במקרה הזה היה די קל, פשוט להפוך את התמונות ונקבל טקסט קריא:

זו הבעיה הראשונה שהתמודדתי איתה, והפתרון שלה היה די קל, אבל כעת, הייתי צריך לחשוב על שני דברים בעקבות פתרון הבעיה הזו. ראשית, כיצד אני מחליט באופן עקבי עבור כל התמונות מה נחשב טקסט "יותר טוב" (האם זה הטקסט של התמונה המקורית או הטקסט של התמונה ההפוכה). שנית, כיצד אני מחליט באילו מקרים להפעיל את ההיפוך ובאילו לא (נחזור לשאלה הזו גם עבור המקרים היותר מסובכים שאיתה התמודדתי).

עבור הפרמטר שאיתו אחליט איזה טקסט לקחת מבין התמונה המקורית או ההפוכה, בתחילה הלכתי על בדיקה של איות באמצעות הספרייה pyspellchecker. ניסיתי את זה על כמה תמונות, אבל ראיתי שזה לא עובד טוב, וניסיתי לחשוב על פתרון אחר שיתן תוצאות יותר טובות. ואז נזכרתי שחלק מהתוצאות של הקריאה באמצעות easyocr היא לתת score שמראה כמה המודל בטוח בקריאה שלו. ניסיתי לבדוק את הסקור הממוצע ולקחת את זה בתור קריטריון ההשוואה, וזה עבד טוב הן בתמונות שהצריכו היפוך והן בתמונות שלא, ולכן החלטתי לקחת את הקריטריון הזה בתור קריטריון ההשוואה.

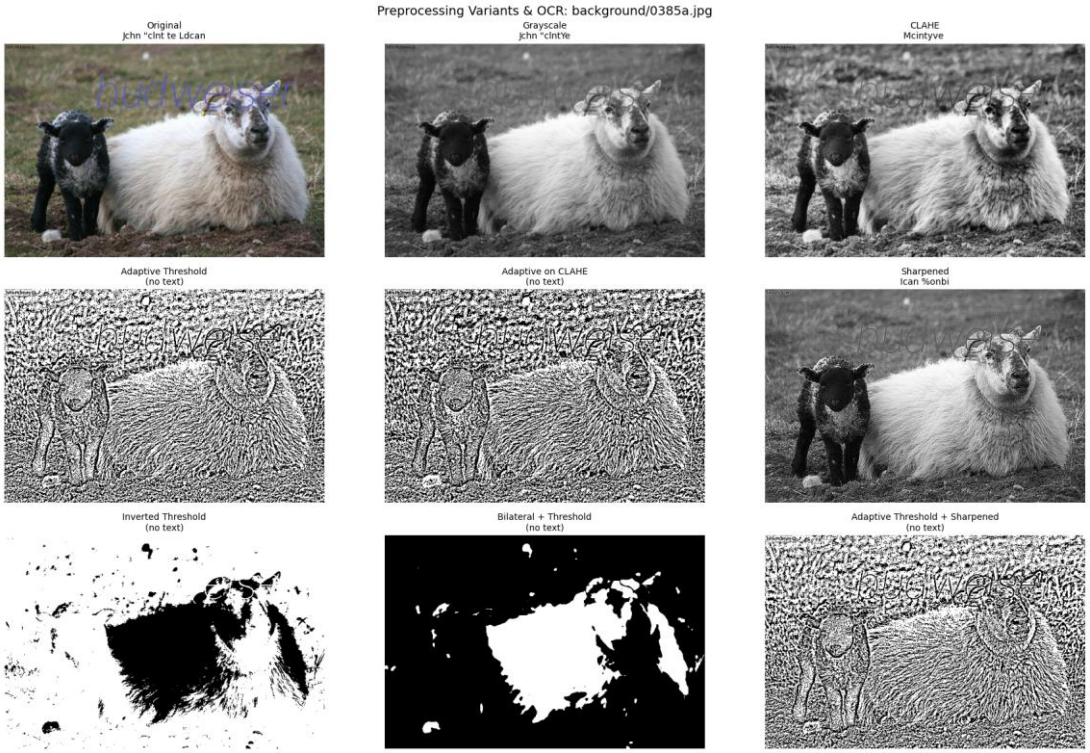
כעת שאלתי את עצמי מתי אדע להפעיל את ההיפוך ומתי לא. התשובה לשאלה הזו הייתה יחסית פשוטה. עבור כל תמונה, אני אבדוק מה הסקור הממוצע על הטקסט שהתקבל עבור התמונה המקורית וההפוכה, ואקח את הטקסט עם הסקור הממוצע הגדול יותר. זה המתודה שהשתמשתי בה גם עבור שאר המניפולציות שאפעיל על התמונות בהמשך, אני מפעיל על כל תמונה את כל המניפולציות שבחרתי להפעיל בסופו של דבר, ובוחר את הטקסט עם הסקור הממוצע הגבוה ביותר.

כדי לבדוק אם התקדמתי, בדקתי את הסקור הממוצע על פני כל התמונות ואת כמות התמונות שהוציאו no_text לפני ואחרי שהפעלתי את המניפולציה של ההיפוך (כאשר עבור כל תמונה אני בודק מה הסקור שהתקבל עבור התמונה המקורית ועבור התמונה המהופכת, ובוחר לקחת את הטקסט מהווריאציה עם הסקור היותר גבוה מבין שניהן). היה ניתן לראות שינוי לטובה הן באחוז התמונות שהפיקו no_text והן בסקור הממוצע של כל תמונה.

```
[Raw Image   ] Avg Score: 0.525, No Text Count: 33 (2.75%)
[Flipped Only] Avg Score: 0.584, No Text Count: 28 (2.33%)
```

כעת עברתי להתמודד עם הבעיה היותר מורכבת של תמונות שהכיתוב בהם דומה לרקע או שהכיתוב לא ברור. הבעיה הזו הייתה יותר מורכבת ובמהלך פתרונה ניסיתי כמה פתרונות עד שהגעתי לפתרון בו בחרתי.

בהתחלה, ניסיתי להפריד את הטקסט עם כמה פילטרים בסיסיים של עיבוד תמונה שלא עבדו בכלל:



לאחר שזה לא עבד, ניסיתי לחשוב בצורה יותר ממוקדת מה הבעיה בתמונה וכיצד אוכל לפתור אותה. שמתי לב שבעיה היא שהכיתוב בצבע דומה מדי לרקע, וחשבתי שאולי אם אגביר את הניגודיות בצורה קיצונית זה יפתור את הבעיה ואוכל לקרוא את הטקסט בצורה ברורה, גם זה לא כל כך עבד:

אמנם לי בתור בן אדם כבר היה יותר קל לקרוא את הכיתוב, אבל זה עדיין לא היה מספיק טוב. מנגד, הכשלון הזה הוביל אותי הלאה. מכיוון ששמתי לב שהכיתוב לפעמים מופיע בצבע דומה לרקע דווקא על ידי הניגודיות, חשבתי שיהיה יותר חכם להפריד את התמונה למרחבי הצבע שלה (R,G,B) ואולי זה מה שיתן לי את הפתרון הרצוי. זה כבר הייתה התקדמות טובה:

אפשר לראות בדוגמה שעבור ערוץ הצבע הכחול, כבר הצלחנו כמעט לקרוא את המילה כמו שצריך.

זה עדיין לא היה פתרון מספיק טוב עבורי, ולכן חשבתי על עוד כיוונים שאוכל להתקדם בהם. מכיוון שראיתי שההפרדה לצבעים עבדה, חשבתי לנסות אולי להעביר את התמונה למרחבי צבע אחרים ואז להפריד לצבעים, ואולי זה יעבוד יותר טוב. ניסיתי להעביר את התמונה למרחב צבע של YUV ולבצע שם הפרדה למרחבי צבע של Y,U,V. זה עבד בצורה הרבה יותר טובה:

אפשר לראות בדוגמה לעיל שעבור הערוץ U קיבלנו את הטקסט הנכון עם סקור של 1.00!

לאורך כל התהליך הזה, בדקתי את הסקור הממוצע עבור כל התמונות ואת כמות התמונות שקיבלו את הטקסט (no_text) שמשמעותו שלא הצלחנו לקרוא כלל. מכיוון שראיתי שישנם מקרים שבהם ההפרדה לצבעים RGB עבדה יותר טוב, **החלטתי לבדוק עבור כל תמונה את כל הווריאציות של היפוך/לא היפוך, הפרדה למרחבי צבע RGB, והפרדה למרחבי צבע YUV. עבור כל וריאציה חישבתי את הסקור הממוצע ואת הטקסט שהתקבל. מבין כל הווריאציות הללו לקחתי את הטקסט של הווריאציה עם הסקור הממוצע הגבוה ביותר מבין כולן בתור הטקסט של אותה התמונה.** כך קיבלתי את התוצאות הבאות עבור כלל התמונות (אנחנו רואים שיפור משמעותי מאוד ביחס להתחלה בשני המדדים, **הסקור הממוצע על פני כל התמונות וכמות התמונות שהוציאו no_text**):

```
[Raw Image   ] Avg Score: 0.525, No Text Count: 33 (2.75%)
[Flipped Only] Avg Score: 0.584, No Text Count: 28 (2.33%)
```

```
[Full Variant] Avg Score: 0.826, No Text Count: 8 (0.67%)
```

כיוון שהגעתי לתוצאות מאוד טובות, החלטתי שזו השיטה שבה אבחר.

להלן כמה מקרים שבהם השיטה לא עבדה, ורעיונות אפשריים לשיפור:

הרעיון הראשון לשיפור שחשבתי עליו, אבל לא היה לי כיוון לרעיון קונקרטי שיחליף את מה שבחרתי בצורה יותר טובה, הוא הפרמטר לפיו אני מחליט איזה טקסט לקחת מבין כל הוריאנטים. אני בחרתי בסקור הממוצע, אבל יתכן מאוד שיש פרמטרים יותר טובים.

כעת אדון במקרים בהן לא הצלחתי לקרוא את הטקסט בצורה טובה, ומה ניתן לשפר.

לדוגמה:

כאן לא הצלחתי לקרוא את הטקסט בכלל, וזה נובע מהעובדה שהצבע של המילים (לבן) והצבע של המסגרת של הטקסט, זהה כמעט לחלוטין. משהו שחשבתי עליו שיכול לפתור את הבעיה אבל לא היה לי זמן להתעסק בו כי זה היה נראה לי מורכב מדי: אולי אפשר היה לנסות לשערך את הכיוונים של המים בתמונה ולפי זה לנסות לחלץ את הטקסט בניקוי של המים. כמו שכתבתי, לא התעמקתי בזה יותר מדי כי כבר לא היה לי זמן.

עוד דוגמה שבה הקריאה לא הצליחה:

כאן הבעיה היא שהרקע דומה מדי לכיתוב, ונראה שטיפלתי בזה, אבל יש מקרים שבהם זה בכל זאת לא עבד, כמו בתמונה הזו. פתרון אפשרי שחשבתי עליו הוא לנסות לחפש מרחבי צבע נוספים שבהם ביחד עם הקצנה של ניגודיות נוכל כן לחלץ את הטקסט בצורה טובה.

<u>עיבוד של הטקסט שהתקבל</u>

לאחר שהגעתי לתוצאות טובות ברמת העיבוד תמונה, עברתי לעבוד על הטקסטים שהתקבלו.

כעת התייחסתי לכמה בעיות שהעליתי לעיל:

1. מילים שמפוצלות למרות שלא אמורות להיות מפוצלות, לדוגמה "wat er" במקום "water".
   או מילים שמחוברות אחת לשנייה.
2. מספרים שמופיעים בטקסט במקום אותיות

3. אותיות קטנות ואותיות גדולות בטקסט.
4. טקסט לא קשור שמופיע בתמונה

נעבור על כל הבעיות ודרך הפתרון שלי:

ביחס לאותיות קטנות ואותיות גדולות בטקסט, **ביצעתי ניקוי של הטקסט:** העברתי את כל הטקסטים להיות עם אותיות קטנות. ביחס למספרים שמופיעים בטקסט במקום אותיות, ביצעתי את המיפוי הבא שמחליף את המספרים להיות אותיות, והפעלתי עוד ניקוי של הטקסט כפי שמופיע בפונקציה שלמטה (פסיקים וסימנים לא קשורים העלמתי, רווחים מרובים הפכתי לרווח בודד). אמנם המיפוי של האותיות והמספרים לא מושלם, כי לדוגמה 1 יכול להיות גם i וגם l, אבל זה פתרון שכן עזר בניקוי והכנה של הטקסט לסיווג.

```python
def clean_text(text):
    """
    Lowercase text, replace digits with letters, remove punctuation, and reduce whitespace.
    """
    text = text.lower()
    num_to_char = {'0': 'o', '1': 'i', '2': 'z', '3': 'e', '4': 'a',
                   '5': 's', '6': 'g', '7': 't', '8': 'b', '9': 'g'}
    for num, char in num_to_char.items():
        text = text.replace(num, char)
    text = re.sub(r'[^\w\s]', ' ', text)
    text = re.sub(r'\s+', ' ', text).strip()
    return text
```

לגבי הבעיה של מילים שמפוצלות למרות שלא אמורות להיות מפוצלות, או מילים שמחוברות אחת לשנייה או טקסט לא קשור שמופיע בתמונה, פתרתי אותה ברמה המסווג (שנגיע אליו בהמשך).

גם כאן המשכתי הלאה בתהליך אחרי שראיתי שהגעתי לתוצאות טובות בשלב הזה.

<u>סיווג הטקסט</u>

כעת עברתי לסיווג הטקסט. המסווג מחולק לשני שלבים: מסווג על בסיס מילים בטקסט ומסווג של רשת NN. ראשית, בהינתן העובדה שהטקסט של האלכוהול מכיל חברות מוכרות של אלכוהול, חשבתי שיהיה חכם ליצור רשימת מילים של חברות אלכוהול, ורשימת מילים של מאכלים או משקאות שאינם אלכוהוליים ולנסות לסווג לפי התאמה של המילים הללו בתוך הטקסט. לשם כך, ביצעתי עוד עיבוד קטן על הטקסטים, ומחקתי את כל הרווחים[1] , וחיפשתי התאמות בתוך הטקסט לחברות האלכוהול או למאכלים ולמשקאות שאינם אלכוהול באמצעות פונקציה של הספרייה rapidfuzz:

```python
ALCOHOL_KEYWORDS = [
    'vodka', 'rum', 'whiskey', 'bourbon', 'tequila', 'cocktail', 'gin', 'smirnoff', 'guinness',
    'absolut', 'heineken', 'brandy', 'stellaartois', 'wine', 'beer', 'lager', 'alcohol',
    'champagne', 'johnniewalker', 'budweiser', 'jackdaniels', 'chardonnay', 'port', 'merlot'
]

NON_ALCOHOL_KEYWORDS = [
    'chocolate', 'hot', 'iced', 'juice', 'espresso', 'cbd', 'coffee', 'herbaltea', 'soda',
    'cannabis', 'weed', 'marijuana', 'sparklingwater', 'sparkling', 'lemonade', 'matcha',
    'icetea', 'icedtea', 'tea', 'water', 'milk', 'milkshake'
]
```

[1] מחיקת הרווחים התבצעה רק עבור השלב הזה ולא עבור הסיווג שהתבצע באמצעות הNN בשלב השני של הסיווג, כפי שנראה בהמשך, שכן ברשת NN שמעבדת טקסטים - <u>כן</u> יש משמעות לרווחים, ואני מעדיף להכיל את הרווחים המוטעים שם כדי לא להרוס לגמרי את כל הרווחים שמשמשים כהפרדה בין טוקנים.

```python
def fuzzy_contains_keyword(text, keywords, threshold=82):
    """
    Check if the text approximately contains any keyword from a list.
    """
    joined = text.replace(" ", "")
    for keyword in keywords:
        if fuzz.partial_ratio(joined, keyword) >= threshold:
            return True
    return False
```

מכיוון שראיתי שישנם מקרים בהם המסווג הזה מחזיר TRUE או False גם עבור מילים של ALCOHOL וגם עבור מילים של NON_ALCOHOL, החלטתי להשתמש בו רק במקרים שהוא מחזיר TRUE עבור אחד מהם בלבד. לאחר מכן, ביצעתי FINE-TUNING של רשת NN מסוג distilbert-base-uncased מאומנת מראש של סיווג על הטקסטים שהיו בידי (פילטרתי את הטקסט עם סקור נמוך מ0.26 באימון של הרשת כדי להימנע מלהכניס רעש בתהליך האימון). כעת, בדקתי את הביצועים של הרשת על סט הtrain וה validation שלי עם שתי וריאציות: ויראציה ראשונה שבה אני משתמש ברשת NN בלבד ללא המסווג שהצגתי לעיל, ווריאציה שנייה שבה אני כן משתמש בו, ורק במקרים שאין לו החלטה ברורה (מחזיר TRUE או FALSE עבור גם עבור מילים של ALCOHOL וגם עבור מילים של NON_ALCOHOL) העברתי את ההחלטה לרשת NN. הבחירה השנייה נתנה תוצאות יותר טובות בכמה אחוזים בכל המדדים עבור סט הולידציה ופגעה רק במעט בתוצאות עבור סט הטריין ולכן החלטתי להשתמש בה. אציין ש**המדדים שבהם השתמשתי כאן היו מדדים שונים של דיוק שבד"כ משתמשים בהם להערכה, והם precision,recall,f1,accuracy.**

אציין עוד שאת הבעיות של מילים שמפוצלות למרות שלא אמורות להיות מפוצלות, או מילים שמחוברות אחת לשנייה או טקסט לא קשור שמופיע בתמונה, שכתבתי שפתרתי ברמת המסווג, אכן פתרתי כאן ברמת המסווג (מילים מפוצלות או מחוברות פתרתי באמצעות זיהוי תבניות של מילים שקשורות לאלכוהול או שקשורות למאכלים או משקאות שאינם אלכוהול, ומילים לא קשורות בכלל באמצעות השימוש ברשת NN שכמו שרואים בתוצאות של המדדים השונים - יודעת לזהות מה רלוונטי מהטקסט ומה לא).

לאחר שהגעתי לתוצאות מאוד טובות ברמת הסיווג, החלטתי שהמסווג ההיבירדי הוא המסווג בו אבחר לסווג את התמונות.

להלן תוצאות הסיווג של התמונות עבור כל אחד מהמקרים הנ"ל (עם ובלי המסווג על בסיס של התאמת מילים):

```
====== NEURAL NETWORK ONLY (TRAIN) ======
              precision    recall  f1-score   support

     alcohol       1.00      1.00      1.00       464
 non_alcohol       1.00      1.00      1.00       469

    accuracy                           1.00       933
   macro avg       1.00      1.00      1.00       933
weighted avg       1.00      1.00      1.00       933


====== HYBRID CLASSIFIER (TRAIN) ======
              precision    recall  f1-score   support

     alcohol       0.98      0.99      0.98       464
 non_alcohol       0.99      0.98      0.99       469

    accuracy                           0.98       933
   macro avg       0.99      0.99      0.98       933
weighted avg       0.99      0.98      0.98       933
```

```
====== NEURAL NETWORK ONLY (VAL) ======
              precision    recall  f1-score   support

     alcohol       0.92      0.93      0.92       116
 non_alcohol       0.93      0.92      0.92       118

    accuracy                           0.92       234
   macro avg       0.92      0.92      0.92       234
weighted avg       0.92      0.92      0.92       234


====== HYBRID CLASSIFIER (VAL) ======
              precision    recall  f1-score   support

     alcohol       0.94      0.97      0.95       116
 non_alcohol       0.97      0.94      0.95       118

    accuracy                           0.95       234
   macro avg       0.95      0.95      0.95       234
weighted avg       0.95      0.95      0.95       234
```

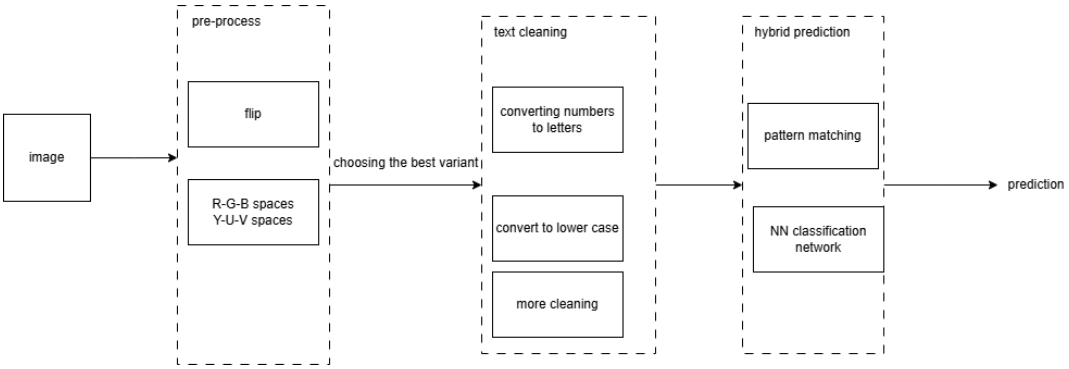
אני מוסיף כאן את הלוגיקה של המסווג ההיברדי למקרה שזה לא היה מספיק ברור מההסבר:

```python
def hybrid_predict(texts, nn_predictions):
    """
    For each text, if fuzzy match clearly indicates alcohol or non-alcohol (but not both),
    return the corresponding class (0 or 1). Otherwise, fallback to neural network prediction.
    """
    hybrid_preds = []
    for text, nn_pred in zip(texts, nn_predictions):
        is_alcohol = fuzzy_contains_keyword(text, ALCOHOL_KEYWORDS)
        is_non_alcohol = fuzzy_contains_keyword(text, NON_ALCOHOL_KEYWORDS)
        if is_alcohol and not is_non_alcohol:
            hybrid_preds.append(0)  # Alcohol
        elif is_non_alcohol and not is_alcohol:
            hybrid_preds.append(1)  # Non-Alcohol
        else:
            hybrid_preds.append(nn_pred)  # In the ambiguous case - use NN
    return hybrid_preds
```

סיכום

במטלה זו עסקתי בניתוח ועיבוד של טקסטים מתוך תמונות וסיווג של תמונות לפי קטגוריות. במהלך התהליך הראיתי את התהליכים שבהם בחרתי לעבוד בעיבוד הדאטה, וכיצד ביצעתי את הסיווג בסופו של דבר.

בסופו של דבר כמו שהצגתי בתחילה, כל תמונה תעבור את התהליך הבא:



כלומר, עבור כל תמונה נבצע היפוך, ועבור התמונה המהופכת והמקורית נבצע הפרדה למרחבי צבע RGB ולמרחבי צבע YUV. נבדוק מה הסקור הממוצע עבור כל אחד מהוריאנטים ונבחר את הוריאנט עם הסקור הגבוה ביותר, ועבורו ניקח את הטקסט שהתקבל עבורו. לאחר מכן נבצע ניקוי על הטקסט כפי שהוסבר לעיל וכפי שמוצג בדיאגרמה, ולבסוף נסווג באמצעות המסווג ההיברידי וניתן את התוצאה הסופית של החיזוי עבור התמונה.