בס"ד

פרויקט למידת מכונה

ID_1: 212555601

ID_2: 212643852

בס"ד

תוכן עניינים

בס"ד

דו"ח פרויקט למידת מכונה

מבוא

פרויקט זה עוסק במשימת סיווג שבה מודל לומד לשייך דוגמאות נתונות למחלקות מוגדרות מראש, בהתבסס על התכונות שלהן. בחרנו לעסוק במשימת סיווג של תמונות מזג אוויר ל5 קטגוריות שונות תוך השוואה מקיפה בין הביצועים של 4 אלגוריתמים שונים שלמדנו.

מטרות הפרויקט הן:

1. להשוות את הביצועים של האלגוריתמים השונים במשימת סיווג תמונות בתחום מזג האוויר.
2. לבדוק איך שיטות להקטנת ממדים משפיעות על הביצועים של המודלים SVM ו RF.
3. לבחון האם שימוש ברשתות נוירונים אכן השיג תוצאות טובות יותר באופן משמעותי בהשוואה למודלים הקודמים.
4. לנתח סוגיות שעלו לנו במהלך הפרויקט.

בפרקים הבאים של הדוח נציג בפירוט את המאגר שבחרנו, תהליך העבודה, האלגוריתמים השונים שהופעלו, תוצאות כל מודל, ניתוח הביצועים, התמודדות עם האתגרים והפתרונות שמצאנו עבורם.

תיאור המאגר

מערך הנתונים נלקח מהמאגר Class Weather Status Image Classification-5 ב Kaggle.
מערך זה כולל 18,039 תמונות המסווגות ל5 מחלקות המתארות תנאי מזג אוויר שונים:

- Sunny, בהיר- 6,274 תמונות
- Cloudy, מעונן- 6,702 תמונות
- Rainy, גשום- 1,927 תמונות
- Snowy, מושלג- 1,875 תמונות
- Foggy, ערפילי- 1,261 תמונות

כל התמונות במאגר הן תמונות צבעוניות מסוג RGB ברזולוציות משתנות שמתארות את המצבים האופייניים לכל אחת מהקטגוריות שהוזכרו.

כדי להכיר טוב יותר את אופי המאגר, ביצענו ניתוח מקדים ובו זיהינו כמה אתגרים עיקריים:

1. חוסר איזון בין המחלקות
המחלקות Sunny ו Cloudy גדולות בהרבה משאר המחלקות Rainy, Snowy ו Foggy וזה מצריך התייחסות מיוחדת בפרויקט שלנו. חוסר האיזון בין המחלקות במאגר נתונים עלול לגרום להטיה של המודל לטובת המחלקות הדומיננטיות יותר. כדי להתמודד עם בעיה זו, שילבנו בתהליך האימון שיטה של חישוב משקלי מחלקות באופן דינמי לפי היחס בין המחלקות. השימוש במשקלים אלו נועד להבטיח שהמודלים יעניקו חשיבות רבה יותר למחלקות בעלות מספר מועט של תמונות וכך לשפר את הביצועים בהן.

2. הבדלי תאורה וזוויות צילום
התמונות במאגר צולמו במגוון תנאי תאורה וזוויות צילום שונות, מה שמגדיל את השונות בתוך המחלקות עצמן. כדי לצמצם את ההשפעה של הבדלי תאורה ביצענו תהליך סטנדרטיזציה ונרמול לתמונות והשתמשנו בשיטה מתקדמת של transfer learning במודל ה CNN.

3. גודל ורזולוציית התמונות
התמונות במאגר הן תמונות צבעוניות (RGB) ברזולוציה משתנה. לצורך תהליך האימון וההערכה של המודלים, ביצענו שינוי גודל לתמונות לגדלים קבועים, באופן הבא:

- עבור KNN, SVM, Random Forest הקטנו את התמונות לגודל אחיד של 64×64 פיקסלים.
- עבור הCNN שמתבסס על ResNet18 השתמשנו בגודל תמונות של 224×224 פיקסלים.

שיטת חישוב המדדים

השיטה שבה השתמשנו לצורך חישוב כל מדדי הביצועים היא macro average.
macro average מחשבת את המדדים בנפרד לכל מחלקה ומבצעת ממוצע פשוט ללא תלות בגודל המחלקות. בחירה זו אפשרה לנו לראות איך המודלים מתפקדים באופן פרטני בכל אחת מהמחלקות.

זיהינו גם את החיסרון האפשרי בשימוש בשיטה זו מכיוון שהיא לא מבטאת את התפלגות הנתונים האמיתית. לכן, בנוסף, הוספנו לשלב האימון (לא לשלב ההערכה) חישוב דינמי של משקלי המחלקות. שיטה זו אפשרה למודלים ללמוד תוך התחשבות באיזון הנתונים בפועל ולשפר את הביצועים גם במחלקות הפחות נפוצות.

<u>שאלות המחקר</u>

בפרויקט זה בחרנו לחקור ולהשוות בין ביצועיהם של ארבעה מודלים שונים במשימת סיווג תמונות מזג אוויר. השאלות המרכזיות שעליהן רצינו לענות במהלך העבודה היו:

1. איזה מודל מבין הארבעה השיג את הביצועים הטובים ביותר בסיווג תמונות מזג אוויר? במסגרת שאלה זו נבצע השוואה מפורטת של ביצועי כל מודל על פי המדדים ונבחן איזה מודל מספק את הביצועים האופטימליים ביותר עבור סיווג התמונות.

2. כיצד משפיעה שיטת הקטנת ממדים (PCA) על איכות הסיווג של SVM וRandom Forest בהשוואה לKNN? כאן רצינו לבדוק האם הפחתת מספר התכונות באמצעות PCA תורמת לשיפור הביצועים של המודלים או דווקא פוגעת בהם והאם יש לה יתרון על פני מודל פשוט כמו KNN שלא מבצע הקטנת ממדים.

3. האם שימוש במשקלי מחלקות באימון המודל CNN אכן משפר את הביצועים במחלקות בעלות מספר נמוך של דוגמאות כמו Rainy, Snowy וFoggy? מטרת שאלה זו היא לבחון האם התאמה של משקלי המחלקות בהתאם להתפלגות הנתונים תורמת לדיוק גבוה יותר במחלקות הקטנות ומהו ההבדל המתקבל לעומת אימון ללא משקלים.

4. האם קיימות קטגוריות שבהן כל המודלים מתקשים במיוחד? ואם כן, למה? ניתוח מטריצות הבלבול מאפשר לנו לזהות מחלקות שבהן שיעור הטעויות גבוה באופן עקבי. בשאלה זו ננסה להבין מה גורם לקושי.

5. כיצד משפיעה העלייה בגודל התמונה על איכות הביצועים בפועל? נבחן את ההבדל בין המודלים שעבדו עם תמונות בגודל קטן יחסית (64×64) לבין CNN שעבד עם קלט ברזולוציה גבוהה יותר (224×224), ונבדוק האם העלות החישובית הגבוהה אכן הובילה לשיפור משמעותי בביצועים.

6. איזה מדד מבין Accuracy, Precision, Recall וF1 הוא החשוב ביותר למשימת הסיווג שלנו? בשאלה זו ננתח מהו המדד שנותן את התמונה המדויקת והאמינה ביותר שבו קיימת התפלגות לא מאוזנת בין המחלקות.

7. מה אחוזי השיפור בין כל מודל ומודל? כדי להעריך את התרומה של כל שלב בפיתוח, נחשב את אחוזי השיפור בין המודלים במונחים של Accuracy וF1 ונבדוק באילו נקודות נרשמה התקדמות משמעותית.

8. האם Augmentation של התמונות משפר ביצועים? Augmentation נועד להעשיר את מגוון הנתונים הזמינים למודל באמצעות שינויים אקראיים בתמונה. בשאלה זו בדקנו האם השימוש בו הביא לשיפור במדדים ובמיוחד בביצועים של המחלקות הקטנות.

מודל הKNN

תיאור האלגוריתם:

מודל זה הוא אלגוריתם סיווג קלאסי ופשוט המבוסס על השוואה ישירה בין הדוגמאות. כאשר המודל מקבל דוגמה חדשה לסיווג, הוא מחשב את המרחק האוקלידי בין דוגמה זו לבין כל אחת מהדוגמאות הקיימות בנתוני האימון. לאחר מכן הוא בוחר את חמש הדוגמאות הקרובות ביותר ומסווג את הדוגמה החדשה על פי המחלקה הנפוצה ביותר בקרב אותן דוגמאות שכנות.

שלבי העבודה:

1. Preprocessing

- כל התמונות הוקטנו לגודל אחיד של 64×64 פיקסלים.
- ערכי הפיקסלים בכל תמונה נורמלו לטווח שבין 0 ל 1.

2. חילוק הנתונים

מערך הנתונים חולק באופן אקראי לשלושה חלקים:

- train (70%)
- validation (10%)
- test (20%)

חלוקה זו מאפשרת הערכה אמינה של ביצועי המודל על נתונים שלא נראו במהלך האימון.

3. אימון המודל

הנתונים הופכים למערכים שטוחים לצורך האימון והמטריצה שהתקבלה מוזנת למודל.
בפרויקט הנוכחי בחרנו להשתמש בחמישה שכנים כלומר k=5 מכיוון שמספר זה הוא נפוץ ומקובל בשביל לאזן בין יכולת ההכללה של המודל לבין הרגישות שלו לרעש ולתנודות בנתונים.

4. הערכת המודל

ביצועי המודל חושבו באמצעות מדדים מבוססי macro, המחשבים את ממוצע מדדי הביצועים עבור כל מחלקה ללא תלות בגודל המחלקה. להלן ההגדרות שהשתמשנו בהן לצורך החישוב:

- TP- מספר המקרים שסווגו נכון במחלקה i.
- TN- מספר המקרים שסווגו נכון במחלקה אחרת.
- FP- מספר המקרים שסווגו בטעות במחלקה i אך שייכים למחלקה אחרת.
- FN- מספר המקרים שזוהו בטעות במחלקה אחרת אך שייכים למחלקה i.

5. מדדי הביצועים

```
KNN model metrics on testing data:
 - Accuracy: 0.4134
 - Precision: 0.4898
 - Recall: 0.4282
 - F1 Score: 0.3631
```

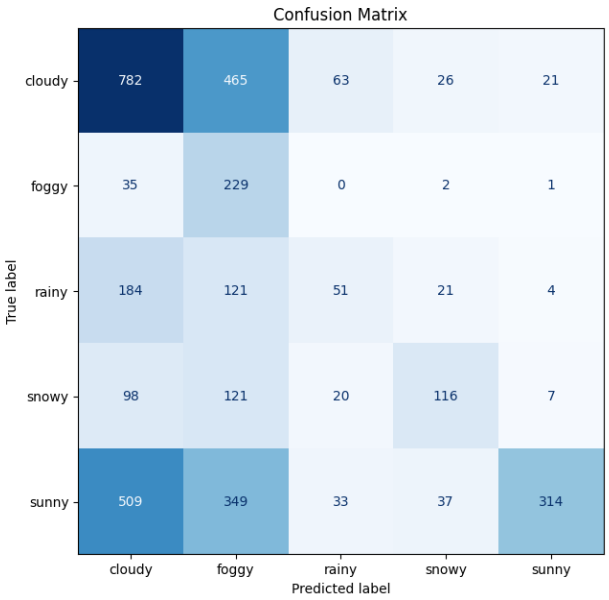
- $Accuracy = \frac{TP+TN}{TOTAL} = \frac{782+229+51+116+314}{3609} = 0.4134$
- $Precision_i = \frac{TP}{TP+FP}$

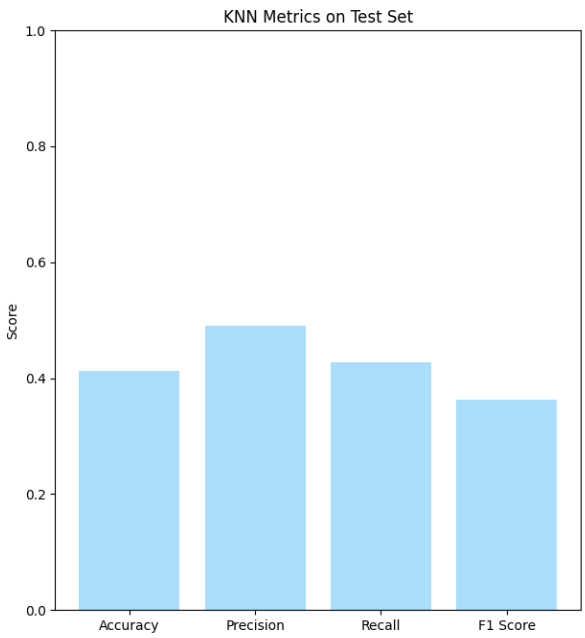
  והPrecision שנקבל עבור המודל: $Macro\ Precision = \sum_{i=1}^{5} \frac{Precision_i}{5} = 0.4898$
- $Recall_i = \frac{TP}{TP+FN}$

  והRecall שנקבל עבור המודל: $Macro\ Recall = \sum_{i=1}^{5} \frac{Recall_i}{5} = 0.4282$
- $F1_i = \frac{Precision\ i * recall\ i}{Precision\ i + recall\ i} X\ 2$

  וה F1 שנקבל עבור המודל: $Macro\ F1 = \sum_{i=1}^{5} \frac{F1_i}{5} = 0.3631$

KNN model





ניתוח confusion matrix

מטריצת הבלבול מצביעה על מספר קשיים משמעותיים בסיווג של KNN.

המודל התבלבל הרבה בין הקטגוריות Cloudy ו Sunny. למשל 509 תמונות ששייכות לSunny סווגו בכלל לקטגוריית Cloudy. קיימות טעויות רבות בין המחלקות הקטנות יותר Rainy, Snowy, Foggy לבין המחלקות הגדולות Sunny, Cloudy, דבר זה מצביע על הקושי של מודל KNN להתמודד עם חוסר איזון משמעותי בין מחלקות.

ניתוח גרף מדדי הביצועים:

ניתן לראות שערכי המדדים נמוכים יחסית, במיוחד הF1 Score , דבר שמעיד על כך שהמודל התקשה מאוד להגיע לסיווג מדויק ואיכותי. הסיבה המרכזית לכך היא הרגישות הגבוהה של KNN לחוסר איזון במחלקות, ובנוסף הרגישות שלו לשונות גבוהה בתמונות עצמן שנובעת מתוך הבדלי תאורה וזוויות צילום.

מודל הSVM

תיאור האלגוריתם:

מודל הSVM הוא אלגוריתם מונחה ללמידת סיווג, שפועל על ידי חיפוש גבול ההפרדה הטוב ביותר בין המחלקות. האלגוריתם שואף למצוא את ההיפר מישור שמפריד בין המחלקות תוך מקסום השוליים בין הדוגמאות הקרובות ביותר של כל מחלקה.
בפרויקט זה השתמשנו במודל SVM עם גרעין מסוג RBF (Radial Basis Function) שהוא סוג של פונקציית דמיון. במקום לנסות למצוא גבול הפרדה לינארי במרחב המקורי של הנתונים, גרעין RBF ממפה את הנתונים למרחב תכונות חדש שבו ייתכן שהנתונים נפרדים בצורה שהיא טובה יותר ואז כך ניתן לסווג גם נתונים שאינם ניתנים להפרדה קווית במרחב המקורי.

בנוסף, בשביל לתת מענה לחוסר האיזון המשמעותי בין המחלקות השתמשנו באפשרות המובנית של המודל לחישוב משקלים אוטומטיים למחלקות. המשקלים מחושבים כך שמחלקות נדירות יקבלו יותר חשיבות באימון והמודל ינסה לאזן בין מחלקות במקום להעדיף את המחלקות הגדולות בלבד.

שלבי העבודה:

1. Preprocessing
   - כל התמונות הוקטנו לגודל אחיד של 64×64 פיקסלים.
   - ערכי הפיקסלים בכל תמונה נורמלו לטווח שבין 0 ל 1.

2. חילוק הנתונים
   מערך הנתונים חולק באופן אקראי לשלושה חלקים:
   - train (70%)
   - validation (10%)
   - test (20%)

   חלוקה זו מאפשרת הערכה אמינה של ביצועי המודל על נתונים שלא נראו במהלך האימון.

3. מיצוי תכונות
   כדי שנוכל לאמן את מודל ה SVM, היה עלינו להמיר את התמונות לפורמט מספרי שניתן לעבדו. כל תמונה בגודל 64×64 ובשלושה ערוצי צבע היא למעשה מטריצה בגודל 64×64×3 = 12,288 ערכים.
   לכן, כל תמונה הומרה לווקטור חד ממדי באורך 12,288 תכונות. תהליך זה שומר את כל המידע הגולמי של הפיקסלים, אבל אינו שומר על המידע המרחבי. המרת התמונה לוקטור שטוח יוצרת אתגר משמעותי של מימד גבוה מאוד וזה עלול לגרום לרעש גבוה וזמן חישוב ארוך.

4. הקטנת ממדים
   כדי להתמודד עם בעיית ריבוי התכונות יישמנו PCA שזה ניתוח רכיבים עיקריים. מטרת השיטה היא לצמצם את ממד הנתונים תוך שימור מירבי של המידע. PCA מזהה את הכיוונים שבהם קיימת השונות הרבה ביותר בנתונים ומקרינה את הדוגמאות על מישור חדש בעל פחות ממדים. במקרה שלנו, צמצמנו את כמות התכונות מ12,288 ל100 רכיבים בלבד וכך למעשה המודל מתאמן מהר יותר. חשוב להדגיש שגם לאחר ההפחתה למאה

רכיבים בלבד, ביצועי המודל היו טובים מאוד וזה מעיד על כך שה PCA הצליח לשמר את עיקר המידע הדרוש לסיווג.

5. אימון המודל

SVM אומן על הנתונים המצומצמים שהופקו מPCA תוך שימוש בRBF אשר מתאים במיוחד לבעיה כמו שלנו שבה הנתונים אינם לינאריים.

6. הערכת המודל

ביצועי המודל חושבו באמצעות מדדים מבוססי macro, המחשבים את ממוצע מדדי הביצועים עבור כל מחלקה ללא תלות בגודל המחלקה. להלן ההגדרות שהשתמשנו בהן לצורך החישוב:

- TP- מספר המקרים שסווגו נכון במחלקה i.
- TN- מספר המקרים שסווגו נכון במחלקה אחרת.
- FP- מספר המקרים שסווגו בטעות במחלקה i אך שייכים למחלקה אחרת.
- FN- מספר המקרים שזוהו בטעות במחלקה אחרת אך שייכים למחלקה i.

7. מדדי הביצועים

```
SVM model metrics on testing data:
 - Accuracy: 0.6312
 - Precision: 0.5681
 - Recall: 0.6378
 - F1 Score: 0.5897
```

- $Accuracy = \frac{TP+TN}{TOTAL} = \frac{741+194+199+220+924}{3609} = 0.6312$

- $Precision_i = \frac{TP}{TP+FP}$

והPrecision שנקבל עבור המודל: $Macro\ Precision = \sum_{i=1}^{5} \frac{Precision_i}{5} = 0.5681$

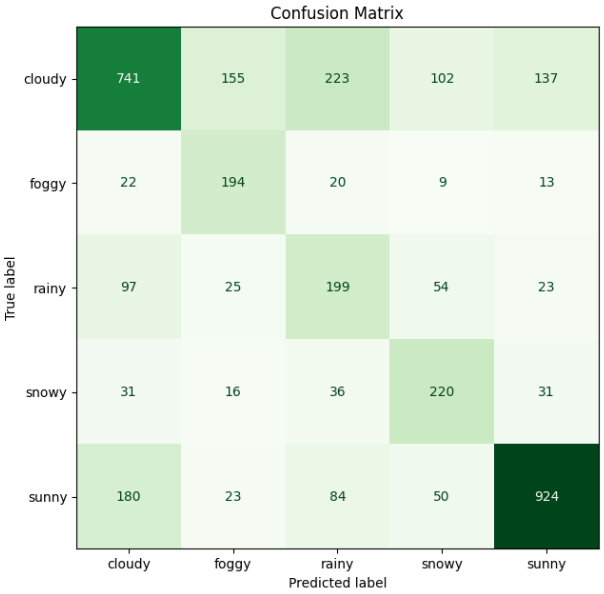
- $Recall_i = \frac{TP}{TP+FN}$

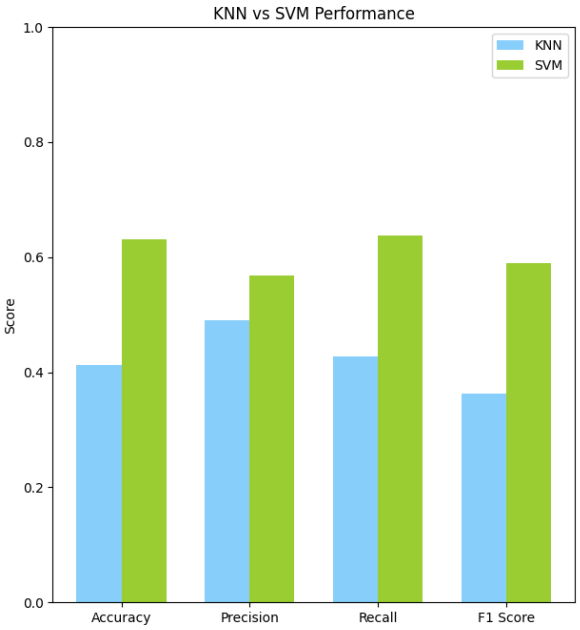
והRecall שנקבל עבור המודל: $Macro\ Recall = \sum_{i=1}^{5} \frac{Recall_i}{5} = 0.6378$

- $F1_i = \frac{Precision\ i * recall\ i}{Precision\ i + recall\ i} X\ 2$

והF1 שנקבל עבור המודל: $Macro\ F1 = \sum_{i=1}^{5} \frac{F1_i}{5} = 0.5897$

SVM model





ניתוח confusion matrix

המטריצה מציגה ביצועים משופרים משמעותית יחסית לKNN. שיפור ניכר בזיהוי של מחלקות כמו Rainy וSnowy, עם ירידה משמעותית בכמות השגיאות לעומת KNN. Sunny סווגה נכון ב924 מקרים, לעומת 314 בלבד בKNN. למרות השיפור, עדיין יש בלבולים בין Cloudy לRainy ובין Cloudy לSunny אך בעוצמה מופחתת.

ניתוח גרף מדדי הביצועים

הגרף ממחיש באופן ברור את השיפור של מודל SVM לעומת KNN בכל מדדי הביצוע.
ההבדלים הבולטים ביותר ניכרים במדדי הRecall והF1 Score מה שמעיד על כך שמודל SVM מצליח לא רק לסווג דוגמאות נכונות בכמות גבוהה יותר אלא גם לזהות בצורה טובה יותר מחלקות נדירות שבמודל KNN כמעט ולא זוהו. שיפור זה מלמד שהשילוב בין שימוש בPCA לבין איזון אוטומטי של משקלי המחלקות מאפשר ל SVM להתגבר על הקשיים שהקשו על KNN.

מודל הRandom Forest

תיאור האלגוריתם:

מודל Random Forest הוא אלגוריתם ensemble שמבוסס על אוסף של עצי החלטה. כל עץ ביער מתאמן על מדגם שונה מתוך הנתונים ותכונות שונות והתווית הסופית נקבעת לפי הצבעת הרוב של כלל העצים. השיטה הזו מסייעת להפחית את הסיכון ל overfitting ולשפר את הדיוק והעמידות של המודל. כדי להתמודד עם חוסר האיזון הברור בין המחלקות במאגר הנתונים השתמשנו באפשרות המובנית של האלגוריתם להתאים את משקלי המחלקות באופן אוטומטי. המשמעות היא שהמודל העניק חשיבות גבוהה יותר לדוגמאות מהמחלקות הנדירות ובכך מנענו העדפה אוטומטית של המחלקות הגדולות.

שלבי העבודה:

1. Preprocessing

- כל התמונות הוקטנו לגודל אחיד של 64×64 פיקסלים.
- ערכי הפיקסלים בכל תמונה נורמלו לטווח שבין 0 ל 1.

2. חילוק הנתונים

מערך הנתונים חולק באופן אקראי לשלושה חלקים:

- train (70%)
- validation (10%)
- test (20%)

חלוקה זו מאפשרת הערכה אמינה של ביצועי המודל על נתונים שלא נראו במהלך האימון.

3. מיצוי תכונות והקטנת ממדים

התהליכים הנ"ל בRandom Forest זהים לחלוטין לתהליכים שנעשו בSVM .

4. אימון המודל

המודל אומן על וקטורי התכונות לאחר PCA. בתוכנית שלנו השתמשנו ב100 עצים ובאופציה האוטומטית שמתחשבת במשקלי המחלקה מה שהקנה למודל יכולת להתמודד טוב יותר גם במצב שבו יש פערים בגודלי המחלקות.

5. הערכת המודל

ביצועי המודל חושבו באמצעות מדדים מבוססי macro, המחשבים את ממוצע מדדי הביצועים עבור כל מחלקה ללא תלות בגודל המחלקה. להלן ההגדרות שהשתמשנו בהן לצורך החישוב:

- TP- מספר המקרים שסווגו נכון במחלקה i.
- TN- מספר המקרים שסווגו נכון במחלקה אחרת.
- FP- מספר המקרים שסווגו בטעות במחלקה i אך שייכים למחלקה אחרת.
- FN- מספר המקרים שזוהו בטעות במחלקה אחרת אך שייכים למחלקה i.

6. מדדי הביצועים

```
Random Forest model metrics on testing data:
 - Accuracy: 0.6359
 - Precision: 0.6954
 - Recall: 0.4778
 - F1 Score: 0.4997
```

- $Accuracy = \frac{TP+TN}{TOTAL} = \frac{1168+133+6+117+871}{3609} = 0.6359$

- $Precision_i = \frac{TP}{TP+FP}$

והPrecision שנקבל עבור המודל: $Macro\ Precision = \sum_{i=1}^{5} \frac{Precision_i}{5} = 0.6954$

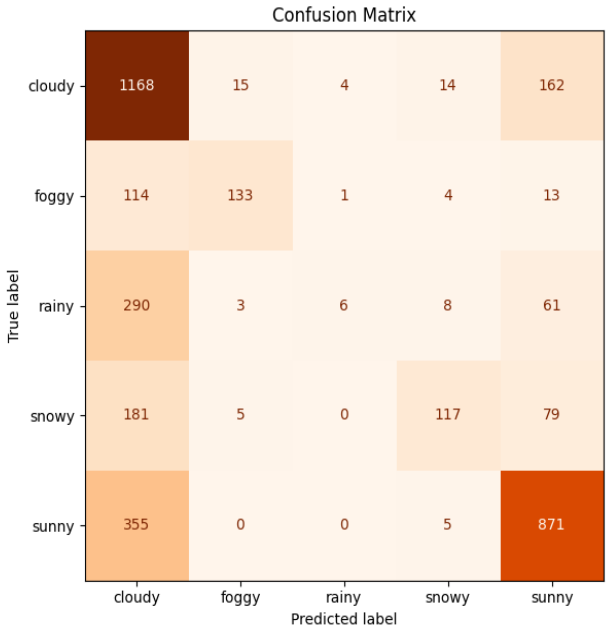
- $Recall_i = \frac{TP}{TP+FN}$

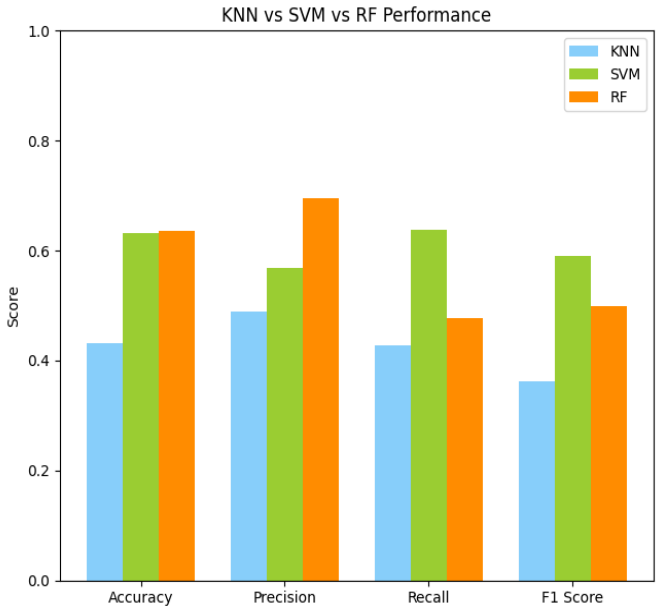
והRecall שנקבל עבור המודל: $Macro\ Recall = \sum_{i=1}^{5} \frac{Recall_i}{5} = 0.4778$

- $F1_i = \frac{Precision\ i * recall\ i}{Precision\ i + recall\ i} X\ 2$

והF1 שנקבל עבור המודל: $Macro\ F1 = \sum_{i=1}^{5} \frac{F1_i}{5} = 0.4997$

Random Forest model





ניתוח confusion matrix

המודל סיווג באופן מצוין את המחלקות cloudy וsunny(1168 ו871 סיווגים נכונים בהתאמה).
יחד עם זאת, רואים שיש הטיה של המודל למחלקת cloudy למשל יש 290 תמונות של rainy, 181 של snowy ו355 של sunny, הדבר הזה מעיד על נטייה של המודל להעדיף את המחלקה הדומיננטית באשר הוא לא בטוח בזה. זה גם מה שמסביר את הRecall הנמוך.

ניתוח גרף מדדי הביצועים:

הגרף מציג באופן ברור את מאפייני הביצוע הייחודיים של מודל RF לעומת KNN וSVM .
המודל מציג את הaccuracy וה precision הגבוהים ביותר מבין שלושת המודלים. נתון זה מעיד על כך שכאשר המודל חוזה שמקרה שייך למחלקה מסוימת הסבירות שהוא צודק גבוהה מאוד.

אבל לצד היתרון הזה, ניתן לראות מגבלה משמעותית. המודל סובל מrecall נמוך יחסית. כלומר הוא מצליח לאתר פחות דוגמאות השייכות בפועל למחלקות הקטנות. הדבר בא לידי ביטוי גם במטריצת הבלבול: מאות תמונות של המחלקות הנדירות יותר כמו foggy, rainy ו snowy סווגו בטעות כcloudy המחלקה הגדולה והנפוצה ביותר. במילים אחרות, המודל מעדיף "לטעות בזהירות" כלומר לנחש את המחלקות הגדולות כשהוא לא בטוח ובכך הוא ממעיט בסיווג חיובי של מחלקות פחות שכיחות. גישה זו אמנם שומרת על דיוק גבוה אבל היא מפחיתה את הרגישות למקרים נדירים ולמעשה יוצרת הטיה חזקה לעבר המחלקות הגדולות.

מודל הCNN

תיאור האלגוריתם:

מודל למידת העברה מבוסס רשת נוירונים קונבולוציונית נבחר לשלב המתקדם ביותר בפרויקט.
המודל מבוסס על הארכיטקטורה של ResNet-18, שהתאמנה מראש על מסד הנתונים ImageNet ומטרתו לספק שיפור משמעותי בביצועים בסיווג תמונות לקטגוריות מזג האוויר.

למידת העברה

למידת העברה (transfer learning) היא שיטה שבה משתמשים במודל שהוכשר מראש על מאגר נתונים כללי כדי לפתור בעיה ממוקדת אחרת. במקרה שלנו, שכבות העומק המוקדמות של ResNet-18 נשמרו "מוקפאות" כלומר, לא עידכנו אותם במהלך האימון כדי לשמר את הידע הכללי על צורות, צבעים ותבניות חזותיות שנרכש באימון הראשוני שכבר קרה.
רק השכבה הסופית של הרשת הוחלפה לשכבת Fully Connected עם 5 יציאות, אחת לכל מחלקת מזג אוויר במאגר שלנו.

שלבי העבודה:

1. Preprocessing
   - כל התמונות הוקטנו לגודל אחיד של 224X224 פיקסלים.
   - ערכי הפיקסלים בכל תמונה נורמלו לפי ממוצע וסטיית התקן של ImageNet.

2. חילוק הנתונים
   מערך הנתונים חולק באופן אקראי לשלושה חלקים:
   - train (70%)
   - validation (10%)
   - test (20%)

   חלוקה זו מאפשרת הערכה אמינה של ביצועי המודל על נתונים שלא נראו במהלך האימון.

3. פונקציית ההפסד
   פונקציית הפסד שנבחרה היא CrossEntropyLoss מכיוון שהיא מתאימה במיוחד לבעיות סיווג רב מחלקתי. היא משקללת את מידת ההתאמה בין הפלטים החזויים של המודל לבין התוויות האמיתיות של הנתונים. הפונקציה מחשבת את האובדן עבור כל דוגמה במערך ומחזירה את הממוצע הכולל. נשים לב שלא התחשבנו במשקלי המחלקות מכיוון שהשכבות ה"מוקפאות" שלו כבר למדו תכונות כלליות של תמונות ממחלקות מאוזנות ולכן נוסחת פונקציית ההפסד נראית כך:

$$\text{CrossEntropyLoss} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{5} y_{ij}\log(\hat{y}_{ij})$$

N = מספר הדוגמאות הכולל.

y_{ij} = התווית האמיתית עבור הדוגמה i והמחלקה j. y_{ij} = 1 אם הדוג' i שייכת למחלקה j, אחרת 0.

$\hat{y}_{ij}$ = ההסתברות החזויה שהדוגמה i שייכת למחלקה j.

4. אופטימיזציה

לתהליך האימון נעשה שימוש באופטימיזטור Adam שמאפשר למידה יציבה ויעילה.
בנוסף שילבנו גם learning rate scheduler שהקטין את קצב הלמידה לאחר מספר אפוקים כדי לאפשר התכנסות עדינה ויציבה יותר של המודל.

5. אימון המודל

אימנו את המודל במשך 10 אפוקים, לאורך האימון עקבנו אחרי מדדי הloss שנמצאו בכל אפוק וראינו שיש ירידה עקבית ב train loss וב validation loss ללא איזשהי חריגה שמעידה לנו על overfitting.

```
Epoch 1/10 - Train Loss: 0.9313, Val Loss: 0.7999
Epoch 2/10 - Train Loss: 0.6674, Val Loss: 0.6349
Epoch 3/10 - Train Loss: 0.6095, Val Loss: 0.5842
Epoch 4/10 - Train Loss: 0.5760, Val Loss: 0.5829
Epoch 5/10 - Train Loss: 0.5622, Val Loss: 0.5537
Epoch 6/10 - Train Loss: 0.5315, Val Loss: 0.5485
Epoch 7/10 - Train Loss: 0.5142, Val Loss: 0.5467
Epoch 8/10 - Train Loss: 0.5195, Val Loss: 0.5527
Epoch 9/10 - Train Loss: 0.5120, Val Loss: 0.5433
Epoch 10/10 - Train Loss: 0.5177, Val Loss: 0.5477
```

6. הערכת המודל

ביצועי המודל חושבו באמצעות מדדים מבוססי macro, המחשבים את ממוצע מדדי הביצועים עבור כל מחלקה ללא תלות בגודל המחלקה. להלן ההגדרות שהשתמשנו בהן לצורך החישוב:

- TP- מספר המקרים שסווגו נכון במחלקה i.
- TN- מספר המקרים שסווגו נכון במחלקה אחרת.
- FP- מספר המקרים שסווגו בטעות במחלקה i אך שייכים למחלקה אחרת.
- FN- מספר המקרים שזוהו בטעות במחלקה אחרת אך שייכים למחלקה i.

7. מדדי הביצועים

```
CNN model metrics on testing data:
 - Accuracy: 0.7745
 - Precision: 0.7621
 - Recall: 0.7933
 - F1 Score: 0.7760
```

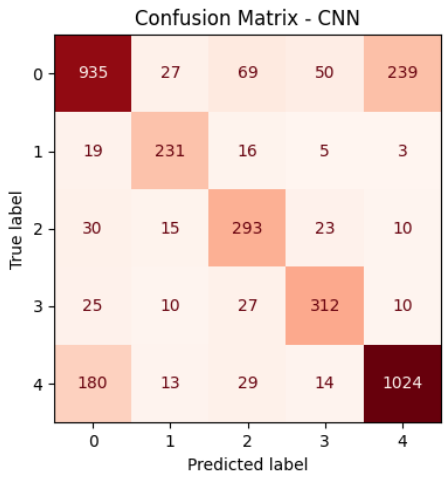
- $Accuracy = \frac{TP+TN}{TOTAL} = \frac{925+231+293+312+1024}{3609} = 0.7745$
- $Precision_i = \frac{TP}{TP+FP}$

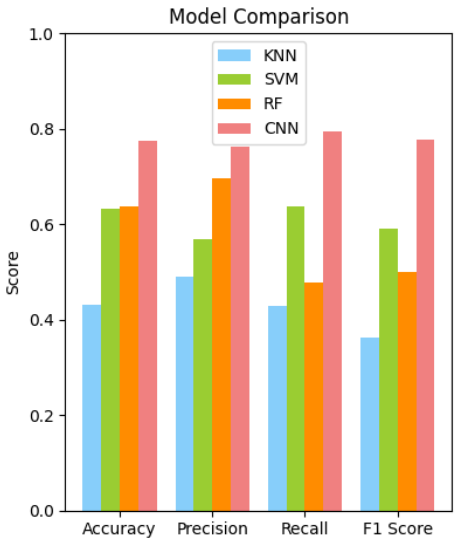
  והPrecision שנקבל עבור המודל: $Macro\ Precision = \sum_{i=1}^{5} \frac{Precision_i}{5} = 0.7621$
- $Recall_i = \frac{TP}{TP+FN}$

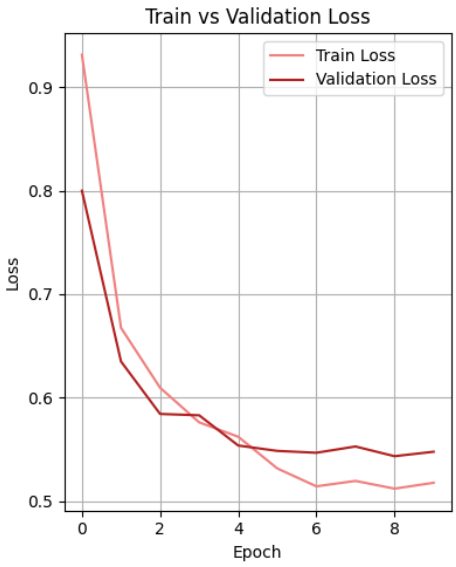
  והRecall שנקבל עבור המודל: $Macro\ Recall = \sum_{i=1}^{5} \frac{Recall_i}{5} = 0.7933$
- $F1_i = \frac{Precision\ i * recall\ i}{Precision\ i + recall\ i} X\ 2$

  והF1 שנקבל עבור המודל: $Macro\ F1 = \sum_{i=1}^{5} \frac{F1_i}{5} = 0.7760$

CNN model







ניתוח confusion matrix

מחלקות כמו Sunny ו Snowy מסווגות באופן מדויק מאוד עם מספר גבוה של True Positives. גם המחלקות הבעייתיות בעבר, כמו Rainy ו Foggy, הראו שיפור מובהק בזיהוי בהשוואה למודלים הקודמים. הבלבול בין מחלקות כמו Cloudy ו Sunny עדיין קיים אבל הוא מצומצם יותר.

ניתוח גרף מדדי הביצועים

אפשר לראות בבירור שמודל הCNN מוביל בכל אחד מהמדדים. הפער בF1 Score לעומת שאר המודלים משמעותי מאוד, מה שמעיד על איזון מצוין בין Precision לRecall. שיפור הRecall בCNN ביחס לRandom Forest מראה שהמודל פחות שמרני ולא מהסס לנבא גם מחלקות פחות שכיחות מבלי לפגוע בAccuracy.

גרף הירידה

גרף הLoss לאורך 10 האפוקים מציג ירידה מאוזנת ואין סימנים ל Overfitting. הScheduler תרם להתייצבות תהליך הלמידה בשלבים המאוחרים יותר. נקודת האיזון של ה Validation Loss הושגה לקראת סוף האימון והמודל לא איבד מיכולת ההכללה שלו.

פתרונות לשאלות

1. איזה מודל מבין הארבעה השיג את הביצועים הטובים ביותר בסיווג תמונות מזג אוויר?

המודל שהשיג את הביצועים הגבוהים ביותר בכל המדדים שנבדקו הוא מודל ה CNN:

מודל	Accuracy	Precision	Recall	F1 Score
KNN	0.4314	0.4898	0.4282	0.3631
SVM	0.6312	0.5681	0.6378	0.5897
Random Forest	0.6359	0.6954	0.4778	0.4997
CNN	0.7745	0.7621	0.7933	0.7760

אפשר לראות שהCNN הצליח לא רק לסווג נכון הרבה תמונות אלא גם לזהות טוב יותר את הדוגמאות מהמחלקות הקטנות מבלי לוותר על הדיוק במחלקות הגדולות. התוצאות הגבוהות של המודל נובעות ככל הנראה מכמה סיבות:

- קודם כל, המודל מבוסס על רשת עמוקה (ResNet18) שלמדה תכונות חזותיות כלליות ממסד נתונים עצום (ImageNet), ואנחנו רק "התאמנו" אותה למקרה שלנו.
- השתמשנו בתמונות ברזולוציה גבוהה יחסית של 224×224 ששמרו על יותר פרטים לעומת התמונות הקטנות של המודלים האחרים.
- בנוסף, שילבנו באימון משקלים דינמיים למחלקות, כדי שהרשת תלמד לא רק את המחלקות הגדולות אלא גם תשים לב למחלקות הנדירות כמו Foggy ו Rainy.

כל השילוב הזה איפשר למודל ללמוד בצורה הרבה יותר מדויקת ולהתמודד טוב עם בעיית חוסר האיזון שהייתה באוסף התמונות שלנו.

2. כיצד משפיעה שיטת הקטנת ממדים (PCA) על איכות הסיווג של SVM ו Random Forest בהשוואה ל־KNN?

במהלך הפרויקט בחרנו להשתמש בשיטת הקטנת ממדים (PCA) על המודלים SVM ו Random Forest אבל לא על KNN. הסיבה לכך היא שמודלים כמו SVM ו RF מתמודדים פחות טוב עם מספר תכונות גבוה מאוד ובמקרה שלנו יש מעל 12,000 פיקסלים לתמונה מה שעלול לגרום ללמידת יתר או לחישוביות איטית ולא יציבה. לעומתם KNN מבוסס על השוואה ישירה בין הדוגמאות והפעלת PCA עליו עלולה לפגוע בדיוק שלו. בפועל, התוצאות הראו שהשימוש ב PCA דווקא שיפר משמעותית את היכולות של SVM ו Random Forest בהשוואה לKNN. לכן ניתן להסיק ששימוש נכון בPCA משפר את הלמידה של מודלים שמבוססים על מרחב תכונות בעיקר כשיש כמות עצומה של פיקסלים שלא כולם רלוונטיים. PCA מצליח לזקק את המידע שהכי חשוב מהתמונה וכך עוזר למודלים להתמקד בעיקר.

3. האם שימוש במשקלי מחלקות באימון המודל CNN אכן משפיע על הביצועים במחלקות בעלות מספר נמוך של דוגמאות כמו Rainy, Snowy, Foggy?

CNN בלי משקלי מחלקות:

```
Epoch 1/10 - Train Loss: 0.8966, Val Loss: 0.6884
Epoch 2/10 - Train Loss: 0.6696, Val Loss: 0.6157
Epoch 3/10 - Train Loss: 0.6198, Val Loss: 0.5978
Epoch 4/10 - Train Loss: 0.5922, Val Loss: 0.5795
Epoch 5/10 - Train Loss: 0.5788, Val Loss: 0.5676
Epoch 6/10 - Train Loss: 0.5469, Val Loss: 0.5615
Epoch 7/10 - Train Loss: 0.5435, Val Loss: 0.5605
Epoch 8/10 - Train Loss: 0.5420, Val Loss: 0.5640
Epoch 9/10 - Train Loss: 0.5355, Val Loss: 0.5623
Epoch 10/10 - Train Loss: 0.5364, Val Loss: 0.5701
CNN model metrics on testing data:
 - Accuracy: 0.7825
 - Precision: 0.7935
 - Recall: 0.7792
 - F1 Score: 0.7858
```

CNN עם משקלי מחלקות:

```
Epoch 1/10 - Train Loss: 0.9313, Val Loss: 0.7999
Epoch 2/10 - Train Loss: 0.6674, Val Loss: 0.6349
Epoch 3/10 - Train Loss: 0.6095, Val Loss: 0.5842
Epoch 4/10 - Train Loss: 0.5760, Val Loss: 0.5829
Epoch 5/10 - Train Loss: 0.5622, Val Loss: 0.5537
Epoch 6/10 - Train Loss: 0.5315, Val Loss: 0.5485
Epoch 7/10 - Train Loss: 0.5142, Val Loss: 0.5467
Epoch 8/10 - Train Loss: 0.5195, Val Loss: 0.5527
Epoch 9/10 - Train Loss: 0.5120, Val Loss: 0.5433
Epoch 10/10 - Train Loss: 0.5177, Val Loss: 0.5477
CNN model metrics on testing data:
 - Accuracy: 0.7745
 - Precision: 0.7621
 - Recall: 0.7933
 - F1 Score: 0.7760
```

Accuracy עלה מעט בהרצה ללא משקלים, כלומר התוצאה של כמה דוגמאות סווגו נכון לא בהכרח נפגעה מהחוסר במשקלים. Precision וF1 Score השתפרו קלות ללא משקלים, מה שמעיד על כך שהמודל פחות "נזהר" במחלקות הקטנות ולכן ביצע פחות שגיאות false positive אבל במחיר של recall נמוך יותר.

שימוש במשקלי מחלקות באימון מודל CNN אכן משפר את הביצועים בקטגוריות עם מעט דוגמאות כמו Rainy, Snowy וFoggy. השיפור מתבטא במיוחד בrecall שהוא מדד מרכזי בזיהוי קטגוריות אלו. למרות ירידה קלה במדדים כמו accuracy וprecision, ההטיה הרצויה לעבר המחלקות הקטנות הצליחה והייתה יעילה ולכן בחרנו את CNN עם המשקלים.

4. האם קיימות קטגוריות שבהן כל המודלים מתקשים במיוחד? ואם כן, למה?

כן, ניתוח מטריצות הבלבול של כל ארבעת המודלים מראה בבירור שיש קטגוריה אחת עיקרית שהייתה קשה במיוחד לכולם והיא מחלקת Cloudy. ברוב המקרים, המודלים נוטים לבלבל בינה לבין Sunny מצד אחד ולפעמים גם עם Foggy מצד שני. התוצאה היא שמחלקת Cloudy מקבלת הרבה False Positives וגם False Negatives. זה קורה מאחר ויש דימיון ויזואלי בין המחלקות, תמונות מעוננות לפעמים בהירות כמו תמונות של יום שמשי או כהות כמו ביום ערפילי. כלומר Cloudy יכולה להיראות כמו Sunny או Foggy והמודל מתקשה להבין את ההבדלים כשאין תכונות ברורות שמפרידות. כלומר יש לנו שונות פנימית גבוהה וזה מקשה על המודל ללמוד תבנית ברורה למחלקה הזו לעומת מחלקות כמו Snowy שיש להן מאפיינים מאוד מובהקים כמו נוכחות של שלג לבן. דווקא בגלל שזו מחלקה עם הרבה תמונות המודלים לומדים אותה אבל זה גורם להטיה.

בKNN, כמעט כל המחלקות הנדירות זוהו בתור Cloudy.
בRandom Forest, כמות הניחושים השגויים למחלקה זו גבוהה במיוחד.
גם בCNN, למרות הביצועים הגבוהים יש בלבול מסוים בין Cloudy ל Sunny וFoggy, אם כי ברמה נמוכה יותר.

המסקנה שלנו הייתה ש Cloudy היא מחלקה עם גבולות מטושטשים גם בעין האנושית ולכן טבעי שהמודלים מתקשים בה.

5. כיצד משפיעה העלייה בגודל התמונה על איכות הביצועים בפועל?

במהלך הפרויקט השתמשנו בשתי רמות שונות של גודל תמונה:
עבור KNN, SVM, Random Forest, הקטנו את התמונות לגודל 64×64 פיקסלים.
עבור המודל CNN השתמשנו בתמונות בגודל 224×224 פיקסלים זהו הגודל הסטנדרטי של קלט לרשת ResNet18.

המודל שעבד עם התמונות הגדולות כלומר הCNN השיג תוצאות טובות בהרבה מכל שאר המודלים בכל המדדים. זה קרה ממספר סיבות:

- **שימור פרטים ויזואליים**
  ככל שהתמונה גדולה יותר, כך נשמרים יותר פרטים כמו מבנה העננים, מעבר בין אזורים כהים ובהירים, תבניות עדינות של גשם או ערפל ועוד. במודלים האחרים, הקטנת התמונה פוגעת בפרטים החשובים האלה.
- **שימור המידע המרחבי**
  רשתות כמו CNN יודעות לנצל מידע מרחבי מהתמונה, הן בונות היררכיה של תכונות, מהפשוטות כמו קווים עד למורכבות יותר כמו צורות, מרקמים. הגדלת התמונה מאפשרת להן "לראות" ולהבין את הקשר בין אזורים שונים בתמונה.
- **שלושת המודלים הראשונים לא מנצלים מבנה מרחבי**
  אחרי ההשטחה לוקטור ב KNN/SVM/RF, הולך לאיבוד כל מידע על מיקום הפיקסלים בתמונה ובשילוב עם רזולוציה נמוכה זה מחמיר את המצב.

אז כן, הגדלת התמונה הייתה שיקול נכון ומשמעותי בפרויקט. אמנם זה העלה את הדרישות החישוביות (לקח לנו מעל לשעה וחצי להרצת הCNN) אבל זה שיפר מאוד את איכות הלמידה והדיוק.

6. איזה מדד מבין Accuracy, Precision, Recall וF1 הוא החשוב ביותר במשימת הסיווג שלנו?

במהלך הפרויקט חישבנו את כל ארבעת מדדי הביצועים כדי לקבל תמונה רחבה על איכות המודלים. עם זאת, המדד החשוב ביותר במודל שלנו הוא F1 Score.

F1 משלב בין Precision = כמה התחזיות נכונות לRecall = כמה דוגמאות אמיתיות זוהו. הוא בעצם שומר על איזון בין דיוק לבין יכולת כיסוי וזה בדיוק מה שהיינו צריכות בפרויקט שלנו.

במערך הנתונים שלנו, חלק מהמחלקות היו מאוד קטנות ולכן רק מדד Accuracy לא היה משקף את המציאות, שהרי אפשר היה לקבל Accuracy גבוה גם אם המודל היה מצליח רק ב Sunny ו Cloudy ופשוט מתעלם מהשאר. לעומת זאת, F1 דורש שהמודל יתפקד בצורה טובה בכל המחלקות לא רק בגדולות.

בגלל חוסר האיזון במחלקות ובגלל הרצון שלנו שהמודל יתפקד טוב גם במחלקות הנדירות בחרנו להתמקד ב F1 Score שנותן פתרון שמאזן בין השניים.

7. מה אחוזי השיפור בין כל מודל ומודל?

כדי להמחיש את ההתקדמות בין שלבי הפיתוח השונים בפרויקט, בחרנו להציג בטבלה את אחוזי השיפור בין המודלים. התמקדנו ב Accuracy שמייצג מדד כללי להצלחת המודל וב F1 Score שמאזן בין Precision ל Recall ומתאים במיוחד למשימות עם חוסר איזון במחלקות כמו שלנו.

מודל	Accuracy	Precision	Recall	F1 Score	שיפור מKNN Accuracy / F1
KNN	0.4314	0.4898	0.4282	0.3631	-
SVM	0.6312	0.5681	0.6378	0.5897	+19.98% / +22.66%
Random Forest	0.6359	0.6954	0.4778	0.4997	+20.45% / +13.66%
CNN	0.7745	0.7621	0.7933	0.7760	+34.31% / +41.29%

8. האם Augmentation של התמונות משפר ביצועים?

במטרה לבדוק האם שימוש בטכניקות של Augmentation תורם לשיפור ביצועי המודל ביצענו סדרת ניסויים שכללה ארבעה שלבים שונים. בכל שלב, שילבנו רכיבים נוספים כמו Transfer Learning או Augmentation ובדקנו כיצד הם משפיעים על דיוק הסיווג ומדדי הביצועים של המודל.

שלב	סוג המודל	Accuracy	Precision	Recall	F1 Score
1	רשת נוירונים פשוטה	0.7143	0.6976	0.6319	0.6570
2	רשת נוירונים פשוטה + Augmentation	0.7498	0.7361	0.6927	0.7105
3	Transfer Learning	0.7745	0.7621	0.7933	0.7760
4	Transfer Learning + Augmentation	0.7348	0.7400	0.7346	0.7349

ניתוח השלבים:

שלב 2 → שלב 1: הוספת Augmentation לרשת פשוטה הביאה לשיפור ברור בכל המדדים בפרט במדדי הF1 וAccuracy. תוספת של וריאציות בתמונות כמו סיבוב, שינוי תאורה וכדומה עוזרת לרשת ללמוד טוב יותר.

שלב 3 → שלב 2: המעבר ל Transfer Learning הביא לשיפור הביצועים גם ללא שימוש בAugmentation. המודל מבוסס ResNet18 ידע לזהות תבניות חזותיות מורכבות הודות לאימון המקדים על ImageNet ולכן הצליח לייצר ביצועים טובים בהרבה מהשלבים הקודמים.

שלב 4 → שלב 3: בניגוד לציפיות הוספת Augmentation למודל המעביר לא שיפרה את הביצועים ואף הביאה לירידה קלה בכל המדדים. ייתכן שהדבר נובע מהעובדה שהרשת כבר רוותה מגיוון תבניות ו Augmentation יצר וריאציות שפגעו בתהליך הלמידה.

מסקנה:

Augmentation הוכיח את עצמו ככלי מועיל כאשר עובדים עם רשת פשוטה אך כאשר עושים שימוש ב Transfer Learning תוספת זו אינה חיונית ולעיתים אף מזיקה. לכן המודל שנבחר לשלב הסופי הוא המודל המבוסס על ResNet18 ללא Augmentation שהציג את הביצועים הגבוהים ביותר.

בס"ד

קישור לגיט

https://github.com/OriaDrori/Machine-learning.git