

מסמך מסכם - פרויקט גמר של הקורס רשותות נירוניים ולמידה عمוקה

מגישים: ניב אלכס (322822602), עדן אליהו (20801489), סיגל שחק (318955549)

נושא: ביצוע Multiclass and Multilabel Classification על תמונות של פוקימונים

תקציר:

בפרויקט הגמר של הקורס רשותות נירוניים ולמידה عمוקה ביצענו שתי משימות מרכזיות העוסקות ב- Classification של תמונות פוקימונים.

חלק ראשון – משימה של Multiclass classification על פוקימונים מדור 1

במשימה הראשונה עבדנו עם DATASET מסוון באתר [Kaggle](#) הכילול כ- 17,000 תמונות מתיוגות של 150 פוקימונים מדור 1. המטרה המרכזית שלנו במשימה הראשונה הייתה לבצע משימה של Multiclass classification הנקראת Classificaion. תמונה חדשה של פוקימון (דור 1) שהמודול עוד לא ראה, לחזות לאיזה class הפוקימון זהה שיר' (במקרה זה יש 150 מחלקות שונות שמייצגות את מספר הפוקימונים השונים הקיימים בדור 1). מטרת העבודה הייתה לשחרר תוצאות בסיס (Baseline), לבצע תהליכי חקירה ושיפור של מודלים כולל הטעמה של עקרונות מרכזיים שנלמדו לאורק הקורס כגון: Transfer Learning, התמודדות עם אטגר' Overfitting וחוסר איזון בין מחלקות, ולנתח את ביצועי המודול על קבוצת Test באמצעות מדר' דיקס Matrix, Confusion Matrix, וניתוח אטגרים וחסמים.

תהליכי העבודה כללו מספר שלבים: תחיליה ביצענו בדיקות תקינות ואיכות של הנתונים הכולר זיהוי תמונות פגומות או בלתי קריאות ובדיקות שפויות קטנה להציג דוגמאות של תמונות מtower הדאטה, ולאחר מכן ניתוח התפלגות המחלקות כדי להבין את רמת חוסר האיזון (טווח של עשרות עד מאות תמונות למחלקה).

לאחר מכן, הנתונים חולקו לsett אימון,sett ולידציה וsett מבחן ביחס של 80/10/10 בצורה עקבית, ובנינו pipelineיעיל בפייתון באמצעות (tf.data) (batch/prefetch) לסייע מהירות האימון ויציבותו.

חלק מההתמודדות עם חוסר איזון במחלקות הוסיף דוגמאות אימון חדשות באמצעות Offline Augmentation יצרת תמונות חדשות ושמירתןsett האימון בלבד, עד ליעד אחד יותר של תמונות למחלקה. ישמןנו גם שיטות ללחימה ב- Overfitting callbacks例如 Dropout, ReduceLROnPlateau ועוד EarlyStopping, ModelCheckpoint לשימרת המודול הטוב ביותר על בסיסsett הווילדייה.

בשלב ייצור המודלים נבנתה רשת CNN Baseline קטנה לצורך השוואה, ולאחר מכן הרכזו מודלים מתקדמים המבוססים על עקרונות של למידת העברה (Transfer Learning) מארכיטקטורות מוכרות שהכרנו במהלך הקורס כגון MobileNet V3-EfficientNet ו- callbacks fine-tuning לשכבות האחוריות במידת הצורך, יחד עם כיוון smoothing, learning rate ו-Hyperparameters.

כל הרצה تعدה בטבלת ניסויים הכוללת שם ריצה, זמן ריצה ותוצאות Test/Val לצורכי ביצוע השוואת מקיפה בין המודלים השונים.

הערכת המודל בוצעה על קבוצת Test בלבד בסיסם התהילה, והוצגו גם גרפים של Accuracy/Loss לאורך האימון, מטריצת בלבול (Confusion Matrix) ניתוח זוגות בלבול נפוצים, וכן דוגמאות של הצלחות וכישלונות של המודל הסופי. המודל המיטב היה מודל EfficientNet fine tune שבוצע עליון smoothing והגיע לדיקן Test של 0.864 כאשר מרבית הטעויות הופיעו בין פוקימונים בעלי דמיון חזותי גבוה (מחלקות קרובות), בעוד שבמחלקות מובחנות המודל הציג זיהוי יציב ובטוח.

חלק שני – ביצוע משימה של Multilabel classification על פוקימונים מדורות 1-7

במשימה השנייה הلقנו צעד אחד קדימה וניסינו לבצע משימה של multi-label classification (multi-label classification – לתחום המשני של פוקימונים). לקחנו תמונות של 809 פוקימונים מאוצר Kaggle ועל בסיסם בנוינו מודלים שאמורים לחזות את הסוג הראשי והסוג המשני של פוקימון חדש. סך הכל קיימים 18 סוגים פוקימונים: רגיל, אש, מים, חשלם, עשב, קרח, לחימה, רעל, אדמה, מעופף, פסיפי, חרק, סלע, רוח, דרקון, אופל, פלאה ופיה. פוקימון ייחד יכול להיות שייך לסוג אחד ראשי בלבד (למשל הפוקימון פיקאצ'י הוא פוקימון מסווג חשלם בלבד) או לסוג ראשי ולסוג משנה (למשל הפוקימון לבאזור שייך לסוג הראשי עשב ולסוג המשני רעל).

מקור הנתונים שלנו למשימה השנייה היה מאגר נתונים של תמונות פוקימונים לאתר Kaggle הכלול **809 תמונות של פוקימונים מדור 1 ועד דור 7**, כאשר כל תמונה של פוקימון עברה עיבוד מקדים שלנו ונקבעה לגודל אחיד של 120 פיקסלים על 120 פיקסלים בפורמט RGB לצורך אימון המודלים.

בנוסף ל-809 התמונות, מאגר הנתונים שלנו כולל גםקובץ CSV המגדיד מידע טבלאי -labels של סוגי עבור כל פוקימון. קובץ CSV מכיל 809 רשומות של פוקימונים ומאפשר לנו לבצע התאמת עקבית בין כל תמונה לבין התוויות (תוויות כלומר הראשי של הפוקימון והסוג המשני של הפוקימון אם קיים) המתאימות לה.

הבעיה שאנו ננסה לפתור במשימה זאת מוגדרת כבעיית Multi-Label Classification ממשום שלכל פוקימון קיימת אפשרות להשתתייך לסוג אחד או שני סוגי כל היתר: סוג ראשי (Primary type) ולוויותים גם לסוג משנה (Secondary type).

הגדירה זו מגדירה יותר מבעיית Multi-Class Classification רגילה, שכן המודל שלנו נדרש להפיק בו זמני מסטר תוויות נכונות ולהתמודד עם תלות אפשרית בין labels שונים, ולספק הסתבריות במקום לבחור מחלוקת יחידה מנכחת.

בעוד שעבודות קודמות בתחום עסקו בחיזוי סוג פוקימונים מתוך תמונות, רובות מהן התמקדו בעיקר בחיזוי **הסוג הראשי בלבד**. במשימה זאת לקחנו את האתגר צעד קדימה וניסינו לחזות גם סוג ראשי וגם סוג משנה, ובכך הגדלנו את מורכבות הבעיה.

תהליך העבודה שלנו כלל מספר שלבים. בשלב הראשון ביצענו עיבוד מקדים לתמונות של הפוקימונים (קריאה מהדטה-CSV, שינוי גודל התמונה ונרמול). לאחר מכן המכינו לכל תמונה של פוקימון את ה-label המתאים לה מקובץ CSV כאשר ה-label הוא רשיימת הסוגים (ראשי ומשנה) של הפוקימון הספציפי המדובר. בשלב הבא, עבור כל label ביצענו קידוד לייצוג Hot-Multi-Hot כדי להקלט למודל.

בשלבים הבאים ביצענו ניתוח ראשוני של התפלגות סוגי הפוקימונים, ובנויות מודלים שונים הכוללים רשות CNN קלאסיות ומודלים של למידת העברה (Transfer learning). אתגר מרכזי שציגו במשימה זו הוא חוסר איזון בנתונים: סוגי מסוימים של פוקימונים מופיעים בתדירות גבוהה בהרבה מאשר אחרים, דבר שעלול להטעות את המודל לטובת מחלוקת נפוצות.

התוצאות הסופיות הראו כי שימוש בלמידה העברתית (Transfer Learning) באמצעות MobileNet שיפור את ביצועי המודל ביחס למודל הבסיסי. למרות שהביצועים נותרו מוגבלים, מדד K-Categorical Accuracy הגיע ל-35% ערך הגבואה יותר בהשוואה לבייסליין. נמצא זה מצבע על כך שהייצוגים החזותיים שנלמדו מראש במודל המאומן תורמים לכיצות טוביה יותר לדרגת הקטגוריות הנכונות של הפוקימונים בין התcheinיות המובילות, גם כאשר המשימה מורכבת (דאטה סט קטן יחסית) והנתונים מאתגרים יחסית (תמונה יחידה של כל פוקימון, בניגוד לחלק הראשון של הפרויקט).

חלק א' של הפרוייקט

רקע לבעה הראשונה – בעיית Multiclass Classification

סיווג תמונות של פוקימונים מהוות אתגר מעוניין בתחום הלמידה העמוקה, בעיקר בשל השימוש בין דמיון חזותי לבין מחלקות שונות בין שנות גבואה בטור אותה מחלוקת. מצד אחד, פוקימונים רבים חולקים מאפיינים דומים (צבעים, מבני גוף, עיניים/אוזניים, קווים מתאר ופירלי סגנון ציור), ובמיוחד כאשר מדובר בהתקפות חזותיות של פוקימון או פוקימונים מאומה משפחה חזותית. מצד שני, גם עבור אותו פוקימון, התמונות שונות להשתנות בזורה משמעותית: תנוכה שונה של הפוקימון, זווית צילום שונה של הפוקימון באותה התמונה, רקע שונה, תאורה שונה, סגנון איור שונה (אנימה/קלפים/תלת-ממד/ציור ידני), ולעיתים גם חיתוך חלק של דמות הפוקימון או פשוט רזולציה נמוכה של התמונה.

במשימה הראשונה בפרויקט אנו מתמקדים בעיית **Multiclass Classification** שבה לכל תמונה משוייך **Τιווג יחיד** מתוך מספר רב של מחלקות (150 מחלקות). המטרה היא לאמן מודל למידה عمוקה שמקבל תמונה קלט ומפיק הסתרות לכל אחת מהמחלקות, כך שהמחלקה בעלת ההסתברות הגבוהה ביותר תהיה שם הפוקימון הנכון. בניתוח פורמלי, מדובר בעיה של **لمידה מפוקחת** (Supervised Learning) עם פונקציית הפסד המתאימה לשיווג רב מחלקתי כגון Cross-Entropy ומדד ביצוע מרכזי כגון ממד ה-**Accuracy**.

אתגר המרכזית במשימה הוא לגרום למודל ללמידה **מאפיינים וייזואליים עדינים** שմבדלים בין פוקימונים דומים (לדוגמה הבדלי צורה קטנים, פרופורציות, פרטים כמו זנב/קרניים/דפוסים), תוך שמירה על **יכולת הכללה** (Generalization) טוביה לתמונות חדשות שלא נראו במהלך האימון. לכן נדרש איזון בין מודל בעל יכולת "יצוג גבואה מספיק כדי להבחן בדקדוקית", לבין מגנונים המפחיתים התאמת יתר (Overfitting) ומאפשרים למודל להתייחס לשונות סגנונית ומבנה נתונים בDATA.

פיתוח נתונים ראשוניים ואתגרים בפתרון הבעיה:

מקור הנתונים והמשימה

הנתונים מאורגנים במבנה תיוקית ...jpg.*>pokemon_name</pokemon>: כך שכל תיוקיה מייצגת מחלוקת (Label). מבנה זה מאפשר יצירת אוטומטית של Dataset לפי שמות תיוקיות, ובפרט בנית מיפוי עקיبي id → label המשמש את המודל באימון ובחיזוי. בנוסף, פורמת זה מקל על בדיקות שעשינו במהלך חלק זה של הפרויקט כמו ספירת דוגמאות לכל מחלוקת, והציג דוגמאות וייזואליות לפי קטגוריות.

אתגרים המרכזיים בחלק זה

1. **חווסף איזון בין מחלקות:** מספר התמונות למחלוקת איננו קבוע (בטוח של כ-215-45). מצב זה עלול לגרום לכך שהמודל ילמד טוב יותר מחלוקת עם הרבה דוגמאות, ויתקשה בחלוקת חלשות יותר. בנוסף, מדד Accuracy כללי עלול להסתיר ביצועים נמוכים במחלקות קטנות, ולכן בנוספ' לניתוח עפ' מדד ה-**Confusion Matrix** בלבד הקפדו גם לבצע ניתוח שגיאות באמצעות Accuracy.

2. **Overfitting:** למקרה שיש DATA לא קטן, מדובר ערך דין במספר תמונות מוגבל יחסית ביחס למספר מחלוקות גבוהה (150). רשתות CNN עלולות "לזכור" דוגמאות או להסתמך על מאפיינים לא רלוונטיים (למשל צבע רקע או סגנון איור נפוץ בחלוקת), מה שਮוריד ביצועים על תמונות חדשות. לכן יש צורך בשיטות לביצוע רגולרייזציה כגון: Dropout, EarlyStopping וכדומה.

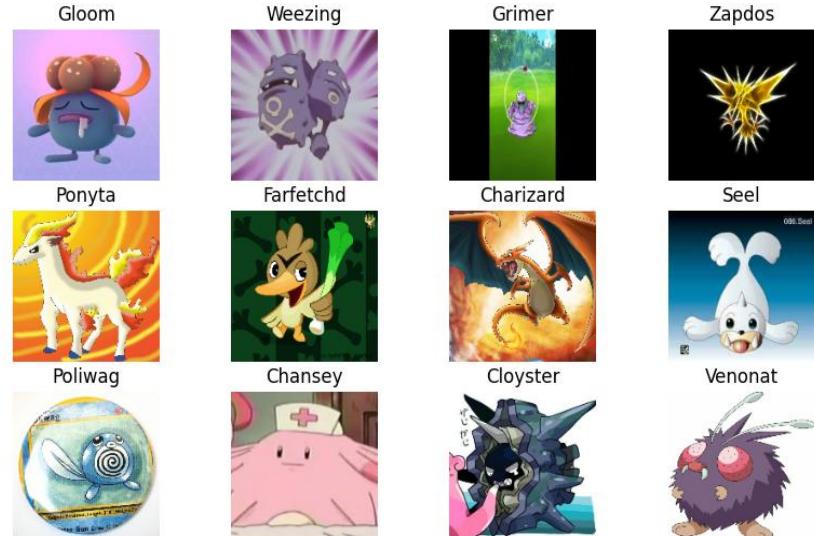
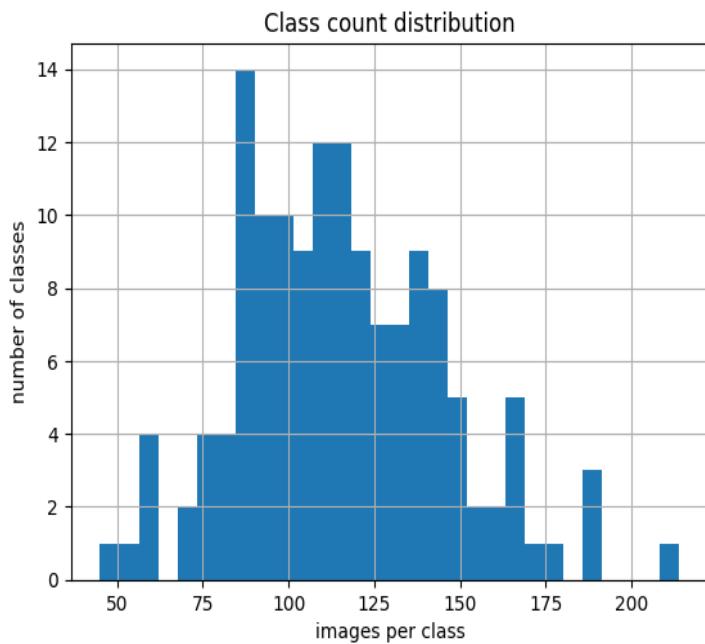
3. **דמיון חזותי:** פוקימונים שונים (ובפרט ההתקפות חזותיות שלהם) יכולים להיראות דומים מאוד, והבדלים ביניהם עשויים להיות קטנים (פרטים בגוף/ראש/זנב, גוונים, פרופורציות). בנוסף, כאשר התמונה חלקיים או באיכות נמוכה, המודל עשוי להתבלבל בין מחלוקות קרובות. לכן הקפדו במהלך חלק זה של הפרויקט גם לבצע ניתוח זוגות בלבד נפוצים ולהציג דוגמאות שגיאות שהמודל חזה.

- 4. תלות בתהיליך הפה-פרוסטיניג:** החלטות כמו גודל התמונה, נרמול התמונה, המראות צבע, ויצוב של משפיעות משמעותית על מהירות האימון, יציבותו והביצועים הסופיים.
- 5. רעש ויזואלי וסגןנות מגוונים:** תמונות עם רקעים עמוסים, טקסט/לוגו, חיתוכים לא מלאים או קונטרסת גמור עשויות להקשות על המודל להתמקד בדמות עצמה. זה גורם שמעלה קושי ומגדיל שונות תוך מחלקטית.

ניתוח נתונים ראשוני

בוצע ניתוח ראשוני כדי להבין את הדאטא ולמנוע שגיאות אימון:

- Audit לתמונות:** נבדקה תקינות של הקבצים, זהו תמונות פגומות הוסרו כדי למנוע קሪסות של המודל שנבנה בהמשך במהלך תהליך האימון שלו.
- התפלגות מחלקות:** בנוינו גרפף היסטוגרמה של מספר התמונות לכל מחלוקת. מצאוו שרוב המחלוקות נעוות סביב 150-80 תמונות, כאשר מחלוקת חלששת יחסית נעוות סביב 45-70 ומעט מחלוקת חזקota מעל 180. תוצאה זו איששה בפנינו את הצורך בהתמודדות עם חוסר איזון למשל באמצעות Offline Augmentation למחלקות החלשות.
- בדיקות דוגמאות:** במחברת הפיתון, לצורך בדיקה שפויות הצגנו batches אקראים מסט האימון בכך לוודא שהתתיוגים נכונים, שמייפוי המחלוקת לשם הפקימון בוצע בצורה נכונה, ושאין טעינה שגואה של תמונות או ריפור צבעים ועיוורים חריגים בתמונות הפקימונים.
- בדיקות עקביות:** ביצענו חלוקה לסט אימון, סט וליידציה וסט מבנן באופן קבוע באמצעות seed ושמרנו את המחלוקת כדי לאפשר השוואת הוגנת בין מודלים והרכזות שונות.



איור 2: דוגמאות לתמונות של פוקימונים מתוך סט האימון

איור 1: התפלגות המחלוקות השונות

הציגת תהליכי החקירה ובנייה הארכיטקטורה:

חלוקת נתונים לשולוש סטים (Train - Validation - Test)

בחalkה זה של הפרויקט ביצענו חלוקה לסטים של Train/Validation/Test ביחס 80% / 10% / 10% בצורה שהיא Stratified כלומר תוך כדי שמירה ככל האפשר על התפלגות המחלקות בכל קבוצה. מטרת החלוקה הייתה לאפשר:

- אימון על סט האימון בלבד.
- בחירה וכיונן של מודלים על סט הווילדייה והשוואות ארכיטקטורות שונות שנבחנו.
- הערכה סופית והוגנת על סט המבחן שנשמר בצד עד סוף התהליך כדי למנוע מצב של overfitting.

כדי לשמר על עיקיות וchooser של התוצאות, הקפכנו לשמר קבצי split קבועים (train.csv, val.csv, test.csv) ולהגדיר seed באופן קבוע. כך, כל הריצה תבצע על אותה חלוקה של נתונים בדיק, וההשוואה בין מודלים תשאיר הוגנת.

עיבוד מקדים (Preprocessing)

כיוון שמדובר בעבודה על מאגר נתונים גדול ולא פשוט, הינו זקנים לייצר כאן סוג של pipeline יעיל שעובר ב责任人 סדרה:

- **Resize לכל התמונות לגודל של 160x160:** בחירה זו מזנת בין זמן אימון לבין שמירה על מידע חזותי מספיק עבור זיהוי של 150 מחלקות.
- **המרה לפורמט של RGB:** המרה זו מאפשרת אחידות במספר הערוצים ומונעת בעיות בקבצים עם פורמטים שונים והכי חשוב, היא מאפשרת קלט סטנדרטי לכל ה-NN שנאנן בהמשך.
- **בדיקות תקינות:** לפני אימון בוצעו בדיקות קריאה לתמונות זהותו תמונות בעייתיות, ולאחר מכן הוצגו akraiim כדי לוודא שתוויות והטינה תקינות batches.

ביצוע אוגמנטציות

כדי להתמודד עם חוסר איזון במחלקות (מחלקות חלשות סביב 45-70 תמונות לעומת מחלקות חזקות), בוצע תהליכי של אוגמנטציה:

- עברו מחלקות עם מעט דוגמאות נוצרו תמונות חדשות ונשמרו לוקאלית, עד ליעד של כ-120 תמונות עברו כל מחלוקת מתוך 150 המחלקות בסט האימון.
- ביצוע אוגמנטציות נעשה באמצעות טרנספורמציות יחסית מתונות על התמונות כגון גזזה, סיבוב, זום קטן ועדיין, שינוי בrama הבהירות של התמונה וכו' וזאת כדי מצד אחד לשמור את זהות הפוקימן אך מהצד השני לייצר וריאציות שונות שלו כדי להפוך את תהליכי האימון לעיל יותר.

היתרון של האוגמנטציות במקורה זה הוא כפוף:

1. איזון DATA אמיתי ברמת התמונות והקבצים.
2. מבחינות שימוש ב-CPUs הגישה הזאת מאפשרת שליטה טוביה יותר ולא מכבדה יחסית על אימון הרשת בזמן אמת אצל הלוקות.

בנייה מודל בסיסי-Baseline CNN model

בנינו מודל Baseline כדי להבין עד כמה קל או קשה לאמן על הדטה שלנו, ולוזא שהשיפורים בהמשך אכן משמעותיים ביחס למודל יחסית בסיסי וסטנדרטי.

1. Small CNN

רשת קונולוציה בסיסית (**Conv** → **Pooling** → **GAP** → **Dense**) שנועדה להיות מודל יחסית פשוט כל, מהיר וմובן. המטרה בבניה של הרשת היזה היא לקבל benchmark ראשון ולהבחן האם הבעה שלנו ניתנת לפתרון טוב ללא טכניקות מתקדמות כגון ביצוע Transfer learning.

2. מודל פשוט יותר להשוואה **Logistic Regression** על Features קפואים
החליטנו לבנות עוד סוג של baseline נוסף שמתבסס על רגרסיה לוגיסטיבית (אותה למדנו בקורס הקודם עם זהה) המודל הזה מפריד בין חילוץ תכונות לבין ה-classifier.

מודלים מתקדמים ולמידת העברה (Transfer Learning)

לאחר יצירת מודל baseline החלטנו לישם את העיקרי של העברה שנלמד בקורס על מודלים שאומנו מראש על הדטה סט של imangenet כדי לנצל תכונות חזותיות כלויות ולהתאים אותן לדטה של פוקימון. המודלים שנבדקו הין:

- **EfficientNetB0** - מודל AMAZON שמספק דיוק טוב ביחס למשאבים.
האימון במשאיות:

1. **Frozen stage**: בסיס המודל היה קפוא הקלספֵיר שלו אומן על ידנו.
2. **Fine tune**: שחרור מהקפהה של חלק מהשכבות האחרונות ואימון בקצב למידה קטן יותר כדי לשפר התאמת לדטה מבלי בעצם לשבור או להרוו את הידע הקודם שהמודל צבר.

• **MobileNetV3Large** - כפי שלמדנו בקורס, המודל של MobileNet הוא מודל יחסית קל ומהיר יותר, מתאים במיוחד לסביבה ללא משאים גדולים (כמו טלפונים שאין להם GPU חזק), הרצה של המודל זהה מאפשרת לנו להשווות tradeoff בין זמן הריצה של המודל לבין הדיוק שלו.

מלחמה ב- Overfitting וכיוונו Hyperparameters

במהלך הרצאות של המודלים החלטנו לנקט במספר צעדים לשיפור הכללה של המודל ולהפחית האוברפיטינג. בין הצעדים העיקריים שבירצנו ניתן למנות את:

- **הגדרה של callbacks לביצוע אימון יציב של המודל**
 - Model Checkpoint: שמירת המודל הטוב ביותר ביותר לפי סט הווילדייזה.
 - Early Stopping: עצירה מוקדמת של הרצת המודל כאשר אין שיפור, למניעת אוברפיטינג וחיסכון בזמן.
 - ReduceLROnPlateau: הורדת learning rate כאשר סט הווילדייזה יחסית תקוע וזאת בכך. לאפשר שיפור של המודל.

- ## • **כינויו של Hyperparameters**

- שינויי של ה-Dropout

- שינוי של קצב הלמידה (Learning rate)

- שינויי מספר epochs בהתאם להתנהגות פונקציית loss-ה ופונקציית ה-accuracy.

- **תיעוד והשוואה בין הרצאות השונות של המודלים**

כל רצחה תועדה בטבלה איחודית שיכללה: שם ריצה, זמן ריציה, best validation accuracy, test accuracy, Validation Accuracy וэтן כדי לראות קיימת התוכנות לערך מסוים והאם יש פער בין Train ל-Test.

הציגות התוצאות:

- נבנתה טבלת השוואה בין הricsות של המודלים שכוללת דיק של כל מודל.
 - המודל המיטב נבחר לפי **Validation Accuracy** ולאחר מכן הוערך על סט המבחן (Test set). (Test set).
 - לבסוף המודל המוביל היה Efficientnet_Finetune שהציג תוצאות של בערך 86% דיק על סט המבחן ו- 84% דיק על סט הולידייביה.

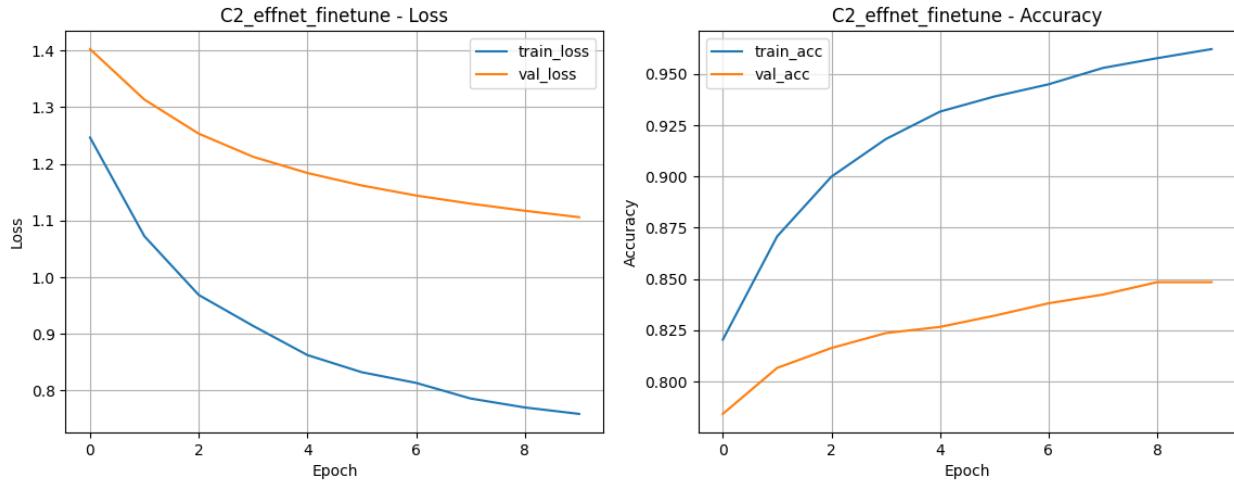
0.848 :Best Validation Accuracy

0.864 :Test Accuracy

	run name	best_val_acc	test_acc
0	Efficientnet_Finetune	0.848	0.864
1	Efficientnet_Frozen	0.840	0.863
2	MobileNet_Frozen	0.827	0.859
3	Lr_on_MobileNet_features	0.795	0.828
4	Small CNN	0.293	0.297

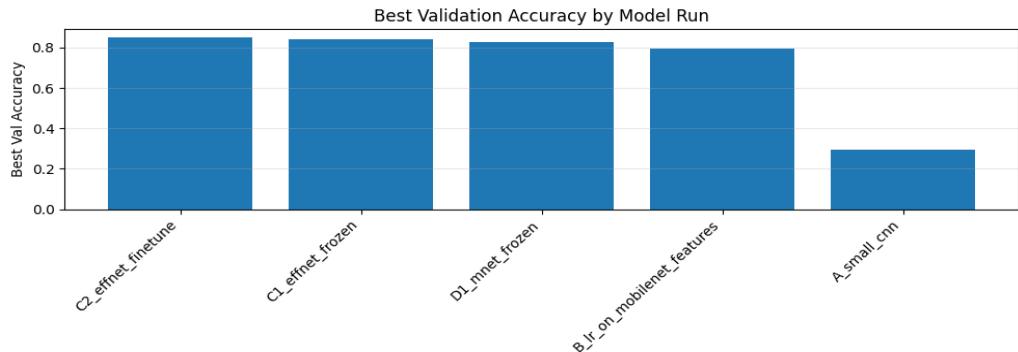
גרפים ויזואלייזציות

גרפי של פונקציית הרפסה (Loss) ופונקציית הדיק (Accuracy) של המודל המדוק ביותר : (Efficientnet_Finetune)

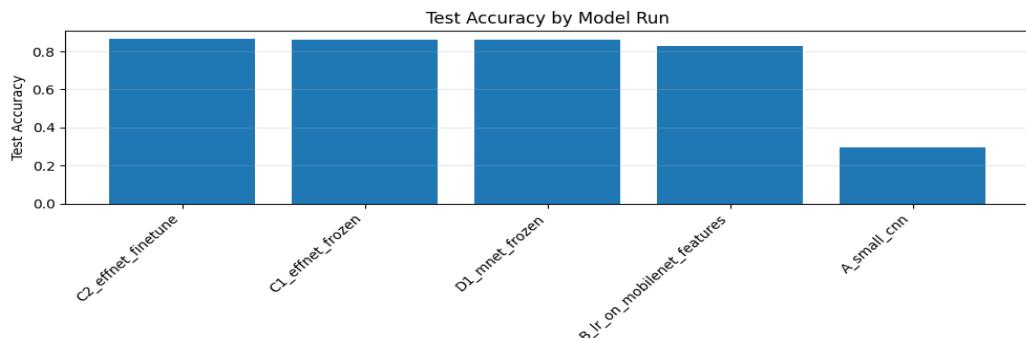


איור 3: תוצאות המודל המדוק ביותר *Efficientnet_Finetune*

גרפי השוואת מודלים :



איור 4: השוואת אחזוי דיק בין המודלים על סט הולידיינה



איור 5: השוואת אחזוי דיק בין המודלים על סט המבחן

דוגמאות הצלחות וכישלונות

כעת לאחר שסימנו את הניסויים שלנו ואת שלב הערכת התוצאות, בחרנו להתקדם עם המודל של Efficientnet_Finetune ותחלנו לבחון אותו על דוגמאות שונות של פוקימונים מסט המבחן.

• 3 תМОנות שהמודל זיהה באופן נכון:



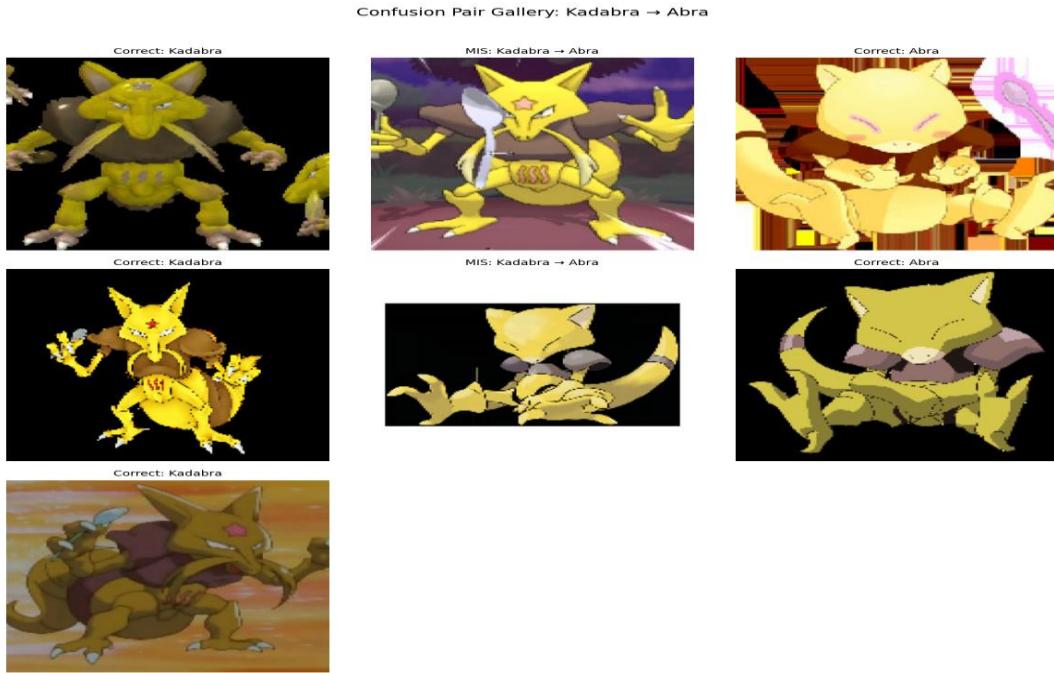
איור 6: תМОנות של פוקימונים שהמודל של Efficientnet_Finetune זיהה באופן נכון

• 3 תМОנות שהמודל זיהה באופן שגוי:



איור 7: תМОנות של פוקימונים שהמודל של Efficientnet_Finetune זיהה באופן שגוי

בשלב הבא הבחנו שלמרות שהמודל יחסית מדויק, הוא לעיתים עדין מתבלבל בין תМОנות דומות של פוקימונים, בעיקר הטעות הראשונה שזיהה פוקימון לא נכון.

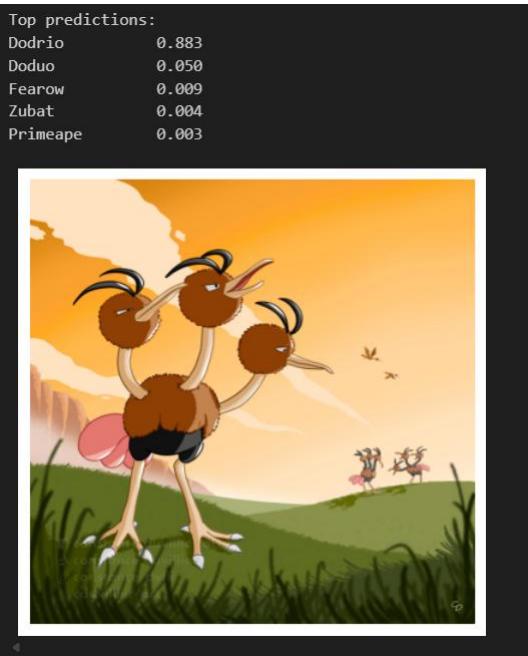


איור 8: דוגמה לבלבול נפוץ של המודל Efficientnet_Finetune, הוא מתבלבל בין הפוקימון Kadabra לפוקימון Abra שהם התפתחות אחת של השני.

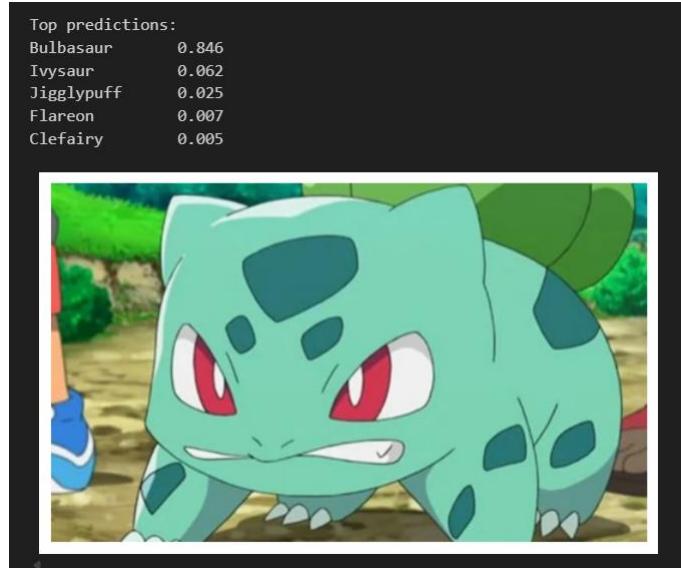
שמירת המודל ובדיקה יכולת הכללה שלו על תמונות חדשות מהאינטרנט שהוא מעולם לא ראה

שמרנו את המודל המאומן כר שnitin יהיה להשתמש בו גם לאחר סיום האימון. לאחר מכן ביצענו בדיקת הכללה (Generalization) על תמונות חדשות שלא הופיעו בدادטה טה המקורי שעליינו עבדנו באתר Kaggle. הורדנו מספר תמונות של פוקימונים מהאינטרנט, העברנו אותן preprocessing לתהיליך resize במסגרתו עשינו לכל תמונה גודל של 160×160 והמרה לפורמת RGB והרכנו את המודל של Efficientnet_Finetune עליו (המודל שנתן את התוצאה הטובה ביותר ביותר).

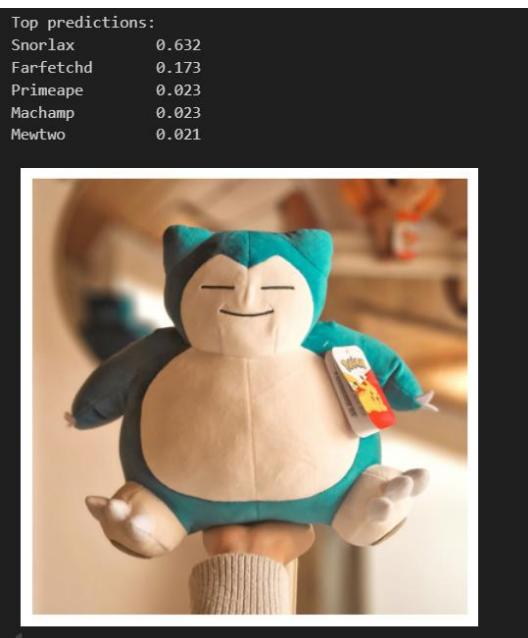
התוצאות הראו שהמודלצליח **לזהות בצורה טובה** גם תמונות חדשות מקורות חיצוניים, למרות הבדלים אפשריים בסגנון ציור, איזוטם תמונה ורקע. ממצא זה מחזק את הטענה שלנו שהמודל לא רק שין את הדאטה טה, אלא גם למד מאפיינים חזותיים כלליים שמאפשרים לו לבצע סיוג מדויק גם על דוגמאות חדשות של פוקימונים.



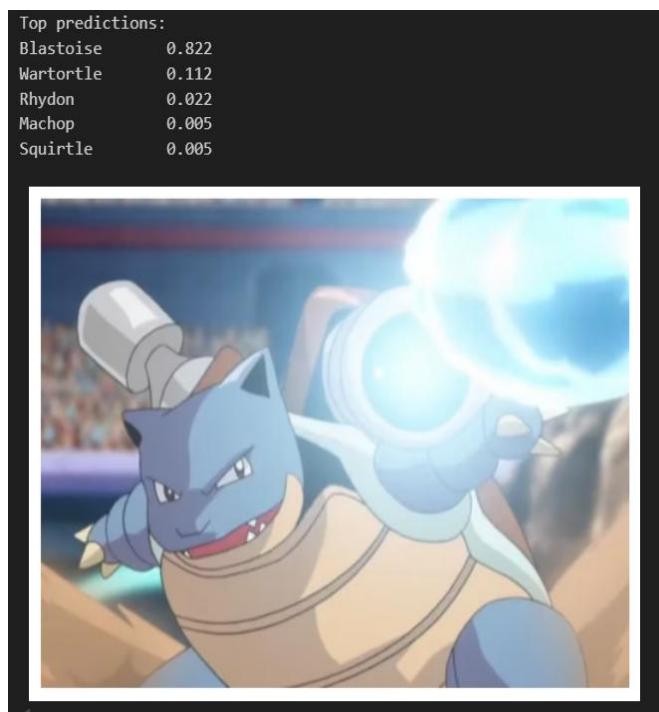
איור 11: תמונה חדשה של הפוקימון דודריין שלקחנו מוגול. מסצנה בסדרה



איור 10: תמונה חדשה של הפוקימון בלבזאור שלקחנו מוגול.



איור 13: תמונה של בובה של הפוקימון סנורלקס, גם כאן המודל מציג ביצועים ייחודיים טובים.



איור 12: תמונה חדשה של הפוקimon בלסטוייז שלקחנו מתוך מסצנה בסרט של פוקימון

חלק ב' של הפרוייקט

רקע לבעה השניה - בעיית Multi-Label Classification

סיווג תמונות (Image Classification) הוא אחד התחומיים המרכזיים ביותר בעולמות של למידה عمוקה ורשתות נוירונים. עם זאת, בעוד שבუיות רבות של סיווג תמונה מנוטה כבעיות Multi-Class (כל תמונה יכולה להיות שייכת למחלקה אחת מתוך מספר מחלקות), קיימות בעיות שבהן כל תמונה יכולה להשתתייר ליותר ממחלקה אחת בו-זמנית. בעיות אלו מוגדרות כבעיות Multi-Label Classification והן נפוצות בעולם האשמי בתחומיים כמו תיוג תמונות (למשל "אדם", "כלב", "רכבת" בהתאם לתמונה) או בניית תמונות רפואיות (תسمינים רבים באוטו צילום רפואי). בפרויקט זה אנו מתמקדים בבעיה מסווג זה: **חיזי סוג פוקימון (Pokemon Types)** מתוך תמונה, כאשר לכל פוקימון יכול להיות **סוג ראשי אחד** ולעתים גם **סוג שני**, כלומר עד שתי תוויות (labels) עבור אותה תמונה.

ביקום של Pokemon קיימים 18 סוגים שונים של פוקימונים (למשל: Fire, Water, Grass וכו') והסוגים של פוקימון משפטים על תכונותיהם, יחסיתו או חולשה בקרבות מול פוקימונים אחרים, ועל האופן שבו שחקנים בונים אסטרטגיה שלהם במשחק. מטרת הפרויקט היא לפתח מודל למידה عمוקה המקבל כקלט תמונה פוקימון ומהזיר כפלט את הסוגים המתאימים לו. בשונה מעבודות רבות אשר מתייחסות לסיווג סוג אחד בלבד (לרבות סוג הראשי), אנו מנסים לבצע חיזוי מלא יותר: גם של סוג ראשי וגם של סוג שני.

במונח למידה סטטיסטית, המשמעות היא שהמודל שלנו אינו נדרש לבחור מחלקה יחידה נcona, אלא להפיק הסתברות לכל אחד מ-18 הסוגים האפשריים, כאשר יותר מסוג אחד יכול להיות נכון.

ניסוח לכך של הבעיה מחייב התאמת הון של ארכיטקטורת המודל והן של פונקציית הפסד שלנו (Loss Function) והمدדים להערכתה, משומש שהנחות בסיסיות של SoftMax אינן תקפות כאשר קיימת אפשרות לשתי תוויות נcona בו-זמנית עבור פוקימון אחד.

אתגרים עיקריים של המשימה הזאת:

1. ביצוע משימה של Multi-Label (2-1 סוגים אפשריים לכל תמונה)

כאשר לכל תמונה עשויים להיות שני סוגים של תוויות, שימוש בפונקציית אקטיבציה מסוג **SoftMax** ובפונקציית הפסד מסוג **Categorical Cross-Entropy** אינו מתאים לפתרון הבעיה. פונקציית SoftMax מניחה שהסתברויות של כל המחלקות מסתכמות ל-100% ושיש מחלקה אחת נcona. בפועל, במקרה שלנו מודל טוב צריך לתת הסתברות גבוהה למספר סוגים במקביל (למשל הסתברות גבוהה ל-Water ו-Flying עבור פוקימון שהוא גם מים וגם מטוף). לכן, פתרון נפוץ הוא שימוש באקטיבציה מסוג **Sigmoid** לכל Label ובפונקציית הפסד מסוג **Binary Cross-Entropy**, או לחלופין לעבוד עם מודל שהוא **Two-Head**.

2. חוסר איזון בנתונים שלנו (Class Imbalance)

אחד האתגרים המשמעותיים שהתמודדנו איתם הוא חוסר איזון בין הסוגים השונים של הפוקימונים - יש סוגים שכיחים מאוד כמו Normal\Water ולוותם סוגים נדירים מאוד כמו Fairy\Dragon\Ice. חוסר איזון זה גורם לכך שהמודל עלול להציג לאורה במשימותו אם ילמד להעדיף סוגים שכיחים, אך בפועל יכשל בדיוני סוגים נדירים - בדיקות אלו שולrob חשוב לנו להזנות נכון. בעיות מסווג זה דורשות שימוש בגישות כמו Weighted Loss, Oversampling וכו'.

3. שונות חזותית בין הפקימונים ותנאי צילום שונים

תמונות פוקימונים עשוות להציג מסגרונות שונות (תמונה בסגנון Artwork, תמונה בסגנון משחק, רקעים שונים, צבעוניות שונה, זוויות שונות). שינוי הרקע או התאורה או הרזולציה עלולה להשפיע על המודל ולהקשות עליו לבצע הכללה נכונה. גם אם הפקימון עצמו מוצג בצורה יחסית נקייה, עדין יש שונות בין תמונות של אותו פוקימון ובין דורות שונים.

4. קביעת threshold נכון להמרת ההסתברויות ל-labels

במודל שמבצע משימה של Multi-Label Classification המודל מhzיר הסתברות לכל סוג, אך בסופן צריך להחליט אילו תווות נבחרות ומוצגות כפלט. זה נעשה בדרך כלל באמצעות threshold (למשל 0.3/0.5), אך אין ערך אחד שתמיד מתאים לכל הסוגים של הפקימונים. סוף גבורה מידי ייצור הרבה False Positives ולבחור K-Top במיוחד כאשר ידוע שכל תמונות של פוקימון Label יחיד או שתי Labels בלבד. במקרה שלנו בחרנו לעבוד עם $k=2$ -Top.

עבודות קודמות שנעשו בנושא:

בתחום סיוג פוקימונים קיימות עבודות ופרויקטים קודמים שמדוגים שימוש ברשתות CNN ובמידת העברה (Transfer Learning) כדי לזרות מאפיינים חזותיים של פוקימונים ולהציג לדיקון גובה יחסית. עם זאת, חלק ניכר מהפתרונותים הקיימים הגיע מתחמקרים בעיה פשוטה יותר מהבעיה שלנו - הם מבצעים ייבוי של type-e/type-k של אותו הפקימון או שהם מבצעים ייבוי של השם של אותו פוקימון לפי התמונה שלו (כמו שהצלחנו לעשות בהצלחה יחסית הרבה בחלק הראשון של הפרויקט). פתרונות כאלה נהנים מבנה פשוט יותר של פונקציית הפסד פונקציית דיקון, ולעתים גם ממערכות נתונים מאוזנים יותר או אלה שמקלים פחות וריאציות של אותה תמונה. החלק הנוכחי של הפרויקט רוצה לחת את הידע שנוצר בנושא עד כה צעד אחד קדימה בכך שהוא מתמודד במפורש האתגר של Multi Label.

ניתוח נתונים ראשוני:

בשלב הניתוח הראשוני ביצמנו היכרות עם מאגר הנתונים של Kaggle-m-CSV, הן בرمת הקטלט הויזואלי (התמונות) והן ברמת התוויות עצמן בקובץ ה-CSV מטרת שלב זה הייתה להבין את מאפייני הדטה, לזהות בעיות פוטנציאליות (כגון חוסר איזון, רשותות חסרות או תמונות פגומות), ומכךן לגזור החלטות בנוגע להמשתתליר העבודה.

קובץ CSV המקורי שmagiyu מה-Kaggle מכיל 809 רשומות ובנוי בצורה הבאה: לכל אחד מ-809 הפקימונים יש عمودה שמצוינת את השם שלו (Name), عمودה שמצוינת את Type1 שלו, عمودה שמצוינת את Type2 שלו וعمודה שמצוינת את ה-Evolution Evolution שלו (פחות רלוונטי לפרוייקט שלנו, لكن הסרנו אותה)

	Name	Type1	Type2	Evolution
0	bulbasaur	Grass	Poison	ivysaur
1	ivysaur	Grass	Poison	venusaur
2	venusaur	Grass	Poison	NaN
3	charmander	Fire	NaN	charmeleon
4	charmeleon	Fire	NaN	charizard

איור 14: נתוני קובץ ה-CSV

בשלב הבא עברנו לדאטה סט שמכיל את 809 התמונות של הפוקימונים וחקרנו קצת עליו. בשלב הראשון הצגנו כמה תמונות לדוגמה כדי לחוש את הדאטה ובשלב הבא וחקרנו על הסוג שלהם (תמונות PNG או JPEG בrcholzieה לא קבועה). ביצענו resize לתמונות (לגדל 120 פיקסלים על 120 פיקסלים) כדי שהמודל יתאמן על תמונות בגודל אחיד.

בשלב הבא יצרנו עמודה חדשה בקובץ CSV שנקראת Type ומכליה את הסוגים של הפוקימון (יכול להיות סוג אחד בלבד יכול להיות שני סוגים). המטרה הייתה שעמודת Type תחליף את העמודות של Type1 ו-Type2.

	Name	Type1	Type2	Evolution	Type
0	bulbasaur	Grass	Poison	ivysaur	[Grass, Poison]
1	ivysaur	Grass	Poison	venusaur	[Grass, Poison]
2	venusaur	Grass	Poison	NaN	[Grass, Poison]
3	charmander	Fire	NaN	charmeleon	[Fire]
4	charmeleon	Fire	NaN	charizard	[Fire]

איור 15: יצירת טבלה חדשה שמכליה עמודה בשם Type

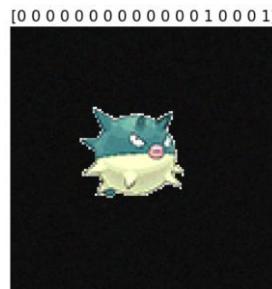
לאחר מכן ביצענו התאמת בין כל תמונה של פוקימון ל-Label של Type שמייצג אותו. בתמונה שלמטה רואים את התוצר הראשוני שקיבלנו, בעת הגענו למצב שלכל פוקימון יש את התווית שמייצגת את הסוגים שלו.



איור 16: התאמת בין תמונה הפוקימון ל-Label שלו

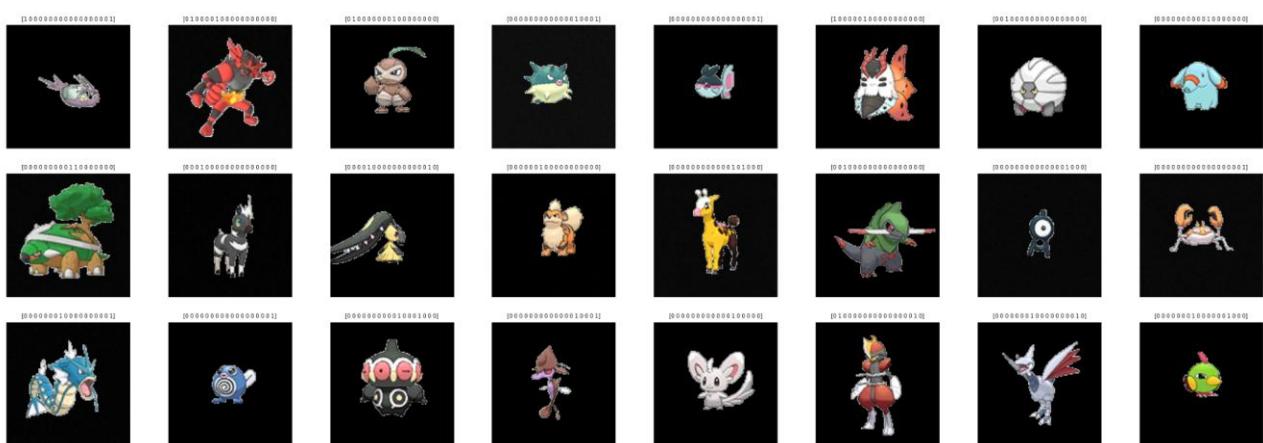
לאחר מכן עברנו לביצוע encoding ל-Label וזאת כדי שלמודל יהיה קלט מnoraml לעבוד איתה. בחרנו לבצע encoding בשיטה של MultiLabelBinarizer עליה חקרנו וגילינו שהיא מתאימה לשימוש מסווג multi-label בהן יש הרבה classes שונות (במקרה שלנו 18 מחלקות שונות).

בהתחליך-h encoding זהה כל Label יהיה מיוצג ע"י וקטור באורך 18 (מספר המחלקות שלנו), כאשר הערך 0 בוקטור יצביע על כך שהפוקימון לא שייך ל-type הנכון והערך 1 יצביע שהפוקימון כן שייך ל-type הנכון. בתמונה למטה ניתן לראות פוקימון שהסוג שלו הוא מים ורעל ואת ה-label המקביל שלו.



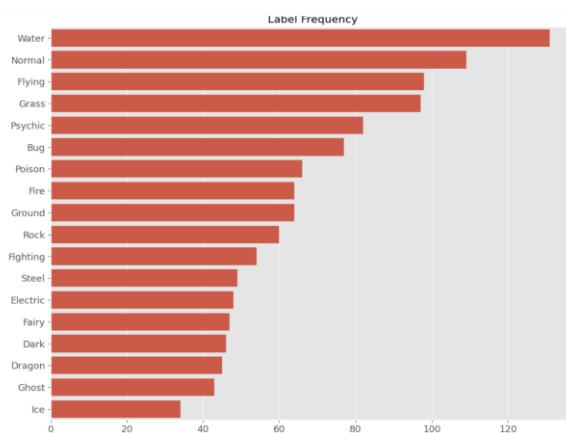
איור 17: ביצוע תחיליך ה-encoding

לבסוף הגענו לתוצר הבא:



איור 18: דוגמאות לדאטה סט הסופי המכיל תמונה ו-label מקודד עבור כל פוקימון

בשלב הבא עברנו לנתח את המחלקות עצמן - הצגנו התפלגות של ה-label השונים של הפוקימונים בדאטה שלנו וראינו שההתפלגות לא אחידה. ניתן לראות שיש יותר מ-120 פוקימונים שהסוג שלהם הוא מים ואילו רק 25 פוקימונים שהסוג שלהם הוא קרח ו-40 פוקימונים שהסוג שלהם הוא רוח. כמובן הדאטה שלנו לא מאוזן.



איור 19: התפלגות הפוקימונים לפי ה-label שלהם

אגומנטציות - לאחר הניתוח הראשוני של הדטה זהינו כי כמות התמונות לאימון המודל יחסית נמוכה (809 תמונות בלבד וזאת עוד לפני שיביצעו חלוקה לסת אימון, סט ולידציה וסט מבחרן). ככלומר כמות התמונות האזינה לאימון המודלים היא יחסית מוגבלת ומאחר שמדובר בשימוש סיווג תמונות, קיימת סכנה של התאמת יתר (overfitting) - מצב שבו המודול לומד מאפיינים ספציפיים מאוד לתמונות האימון ואינו מצליח להכליל לדוגמאות חדשות. על מנת להתמודד עם בעיות אלו, החלטנו לבצע **Data Augmentation** כפי שלמדנו בקורס בשיעור על התמודדות על DATA קטן במודול CNN.

האגומנטציות נעשו להשיג מספר מטרות מרכזיות: בראש ובראשונה הגדלת נפח הדטה – ככלומר יצירתי וריאציות חדשות של כל תמונה קיימת מבייל לאסוסף נתונים נוספים. בנוסף שיפור יכולת הכללה של המודול – כאשר אנחנו מבצעים אוגומנטציות על התמונות המודל נאלץ ללמידה מאפיינים חזותיים כלליים (צורה, צבע, קווי מתאר) ולא להסתמך על זיהוי פרטיים מקריים.

האגומנטציות נבחרו בקפידה כך שייאימו לאופי התמונות (פוק'ימונים עם דמות מרכזית ורקע פשוט יחסית), וימנו משינויים שעולים לשנות את משמעות התמונה או את סוג הפוק'ימון. להלן סוג האוגומנטציות שהכרנו לבצע והציגן שעומד מאחוריה:

- הזזה וביצוע זום עדין – בוצעה הרחבה של גבולות התמונה באמצעות padding ולאחר מכן חיתוך אקראי וזרה לגודל המקורי של התמונה. פועלה זו מדמה שניי קל במיוקם הפוק'ימון בתוך התמונה, ומאלצת את המודל להיות רגיש למיקום המדויק של האובייקט. ההיגיון מאחורי הפעולה הוא שבפועל, פוק'ימון יכול להופיע מעט שמאליה, ימינה או במרכז התמונה, והסוג שלו אינו תלוי במיוקם.
- שניי בהירות וניגודיות – בחרנו לבצע שניי בהירות וניגודיות כללים, בטוחים שמדוברים כשמרניים. ההיגיון מאחורי המהלך היה שמאפייניו קבוע ותאorra עשויים להשנות בין תמונות (או עקב עיבוד שונה), אך סוג הפוק'ימון נשאר זהה. אוגומנטציה זו מונעת מהמודול שלנו להינעל על ערכי קבוע מדויקים.
- שניי קל של ה-saturation – צבעים הם מאפיין חשוב בסיווג סוג פוק'ימונים, אך שניי קל ברוחו של הצבעים (Saturation) לא אמרו לשנות את הסוג. לכן בוצעו שנייים מינוריים בלבד, כדי לא לפגוע במשמעות הסמנטיבית של הצבע.
- הוספת רעש גאושי קל – לתוכנות הוסף רעש אקראי בעוצמה נמוכה. הרצינול מאוחר בבחירה זו היה שרעש קל מאלץ את המודול ללמידה דפוסים יציבים ומשמעותיים ולאאפשר לו להסתמך על פיקסלים בודדים או פרטיים חדים מדי.
- אוגומנטציות שנמענו מהן במקוון – במהלך הפרויקט נמנעו מאוגומנטציות אגרסיביות מדי כגון: טשטוש חזק (Blur) שניי גוון משמעותיים וחיתוכים אגרסיביים שעולים להסרת חלקים מרכזיים מהדמota של הפוק'ימון. אוגומנטציות אלו עלולות לפגוע במידע קריטי לסיווג סוג הפוק'ימון, במיוחד כאשר קבוע לצורה הם מאפיינים דומיננטיים בתחום הזיהוי.

הציגת תהליכי הבדיקה ובניית הארכיטקטורה:

מודל הבסיס:

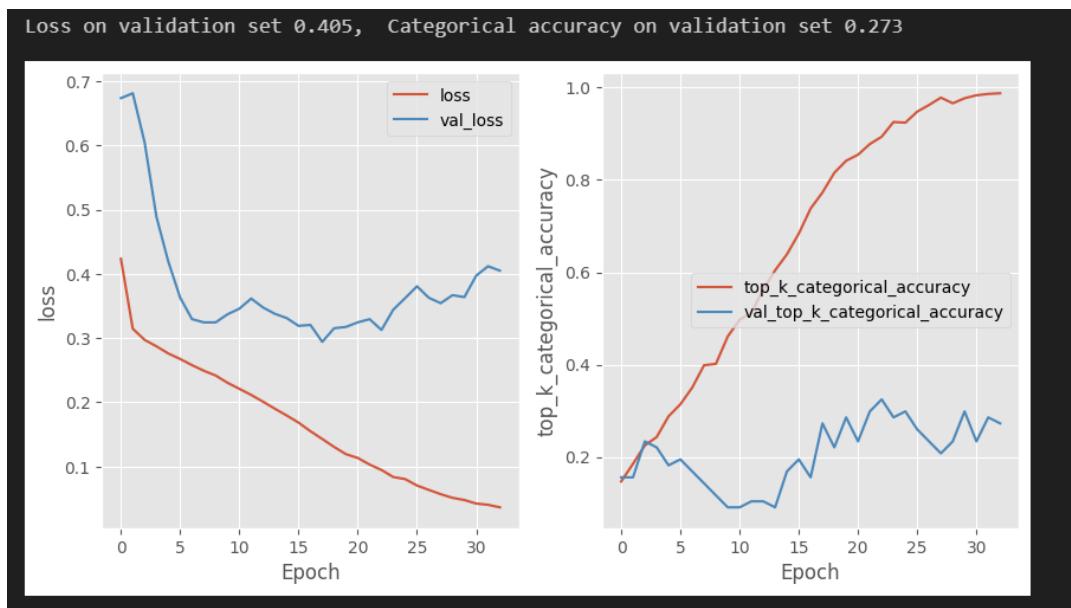
מודל הבסיס שנבנה בפרויקט הוא רשת CNN המיועדת לחילוץ מאפיינים חזותיים מהתמונות של פוק'ימונים ולהרמתם לייצוג שמאפשר סיווג. הארכיטקטורה שבנוינו כוללת חמישה בלוקים קובולוציוניים, כאשר בכל בלוק מופעלת שכבות קובולוציה עם פילטר בגודל 3×3 ופונקציית אקטיבציה מסוג ReLU מספר הפילטרים עולה בהדרגה לאורך הרשת ($150 \rightarrow 128 \rightarrow 64 \rightarrow 32 \rightarrow 16$), כך שהשכבות הראשונות לומדות מאפיינים יחסית בסיסיים של הפוק'ימונים כמו קטנות ומרקמים, והשכבות העמוקות יותר לומדות דפוסים מורכבים של פוק'ימונים ברמת אבסטרקציה גבוהה יותר הקשורים לצורה ולמבנה של הדמות. לאחר כל קובולוציה ביצעונו BatchNormalization לאוצרן י齊וב ונרטול הפלט של השכבה ושיפור יציבות ומהירות האימון, ולאחר מכן ביצעונו MaxPooling בגודל 2×2 ככלומר sampling down של ה-**feature maps** של ה-**feature extraction**, שיטחנו את

הפלט לווקטור חד מימדי (ביצוע Dense(64, ReLU) והעברנו אותו לשכבה flatten. לבסוף מופיעה שכבת פלט בגודל N המפיקה ציון לכל קטגוריה אפשרית של פוקימון ומהווה את בסיס ההחלטה של המודל בסיווג.

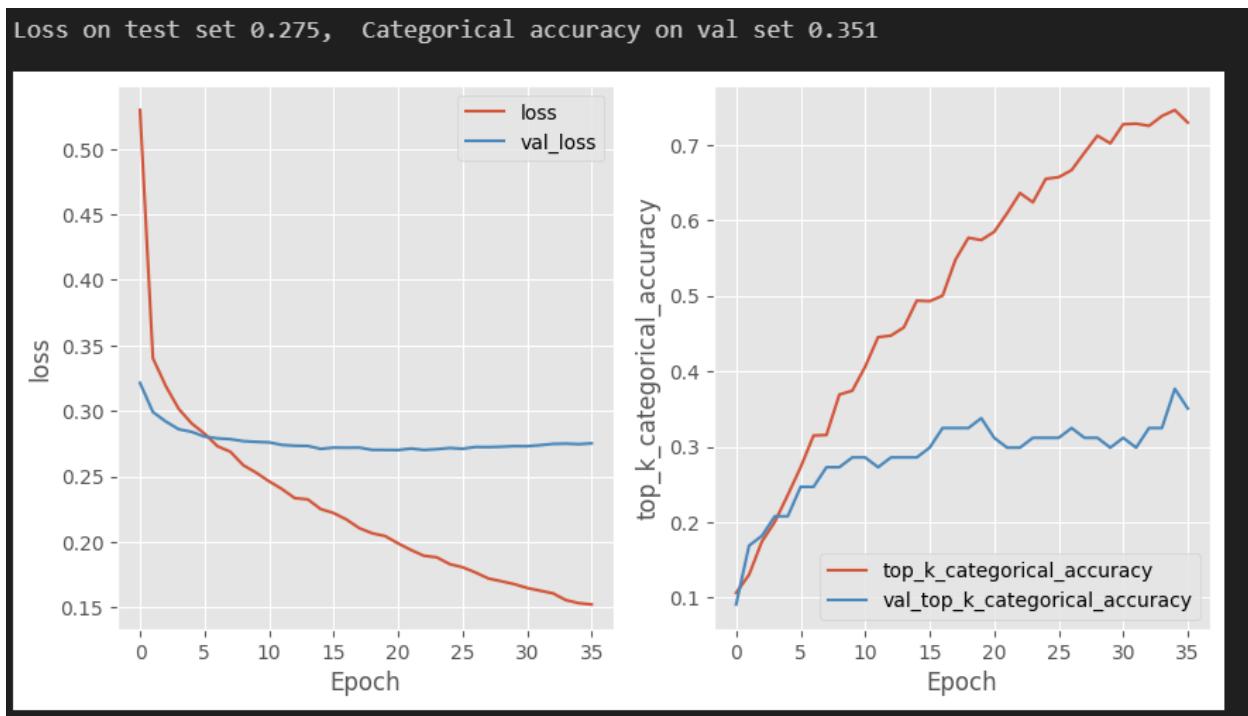
מודל שבסיסו על Transfer Learning ממודל של MobileNet

המודל מבוסס על גישת Transfer Learning באמצעות MobileNet שימוש במרקחה זהה כמודל בסיס מאומן מראש. MobileNet מספק ייצוג חזותי עשיר המבוסס על ידע שנלמד ממאגר גדול כגון ImageNet ובכך מאפשר השגת הכללה טוביה יותר גם כאשר כמות הנתונים בדעתה היא יחסית מוגבלת. מעל ה-backbone של מובייל נט הוספנו **קלאספייר** "יעודי" הכלל שכבת GlobalAveragePooling2D אשר מבצעת ממוצע מרוחבי על ה- feature maps. לאחר מכן נוספה שכבה (Dense(128, ReLU) לצורך התאמת הייצוג הכללי של MobileNet למשימת הסיווג הSPECIFICית אליה אנחנו מתמודדים, וכן שכבת Dropout(0.2) שמבצעת רגוליזציה לשיפור הכללה. לבסוף, שכבת פלט בגודל N מפיקה ציונים עבור כל קטגוריה אפשרית של פוקימונים, אשר הערכת הביצועים מתבצעת על בסיס דירוג התוצאות (K-top) כלומר בדיקה האם התווית הנconaה מופיעה בין K התוצאות בעלות ההסתברות הגבוהה ביותר (במקרה שלנו $K=2$).

הציגת התוצאות:



אייר 20: תוצאות של מודל הבסיס

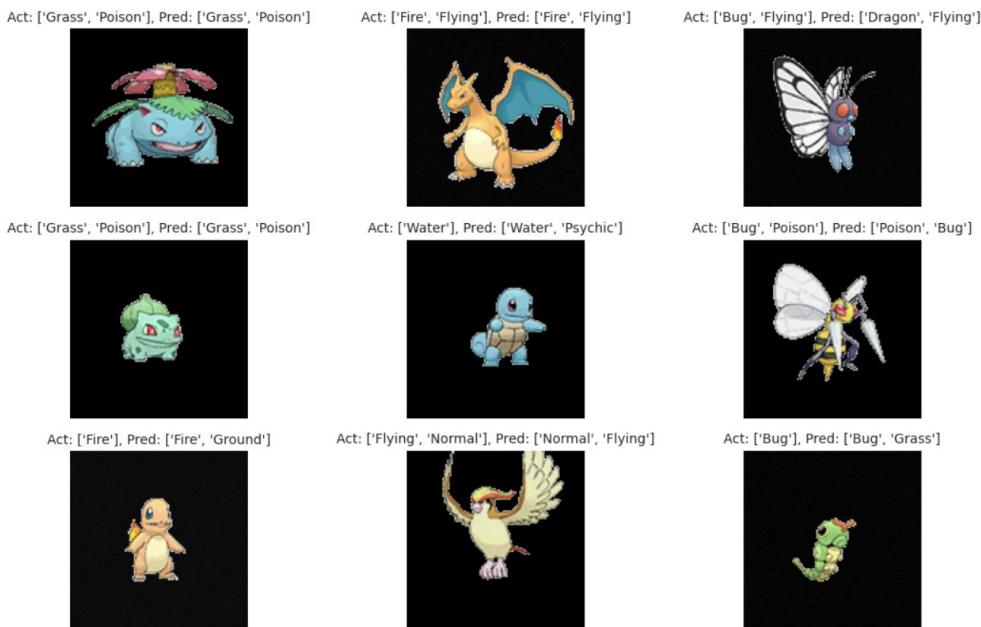


איור 21: תוצאות של מודל המבוסס על MobileNet

מתוצאות עולה כי קיים פער ביצועים מובהק בין מודל-h-CNN הבסיסי שבנוינו לבני המודל המבוסס על MobileNet. במודל הבסיסי נצפית ירידה עקבית ב-train loss המעידת על יכולת המודל להתאים את עצמו לנתחי האימון, אולם במקביל ה-val loss יורד מוקדם יחסית ו אף מתייחל לעלות לאחר מספר epochs. תופעה זו, יחד עם העובדה שה-Top-K categorical accuracy על סט הווילדי ציהו יותר נמוך יחסית (בסביבות 0.25-0.30), מצביעה על נטייה של המודל היסי לבער overfitting ועל הכללה מוגבלת לנתחים חדשים שלא ראה במהלך האימון. לעומת זאת, המודל המבוסס על MobileNet מציג התנהגות יציבה ומשופרת יותר: ה-val loss נמוך ועקביו לאורך האימון, וה-top-k categorical accuracy ווט המבחן מגיע לערך גבוהים יותר (כ-0.35).

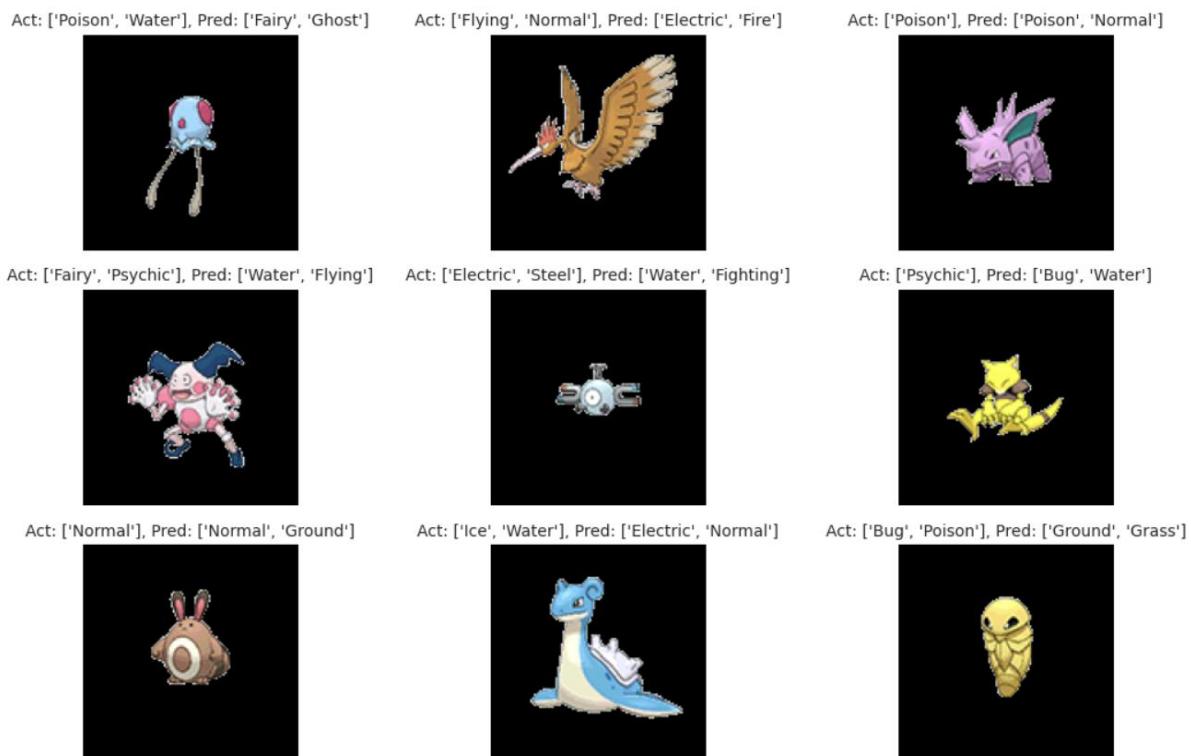
תוצאות אלו מצביעות על כך שהשימוש ב-Transfer Learning מאפשר למודל לנצל ייצוגים חזותיים כלליים שנלמדו מראש, ובכך לשפר את יכולת הדירוג של הקטגוריות הנכונות גם כאשר אין התאמת מושלמת ביןביי הראשוני. בסך הכל, הערה בamusotot Top-K categorical accuracy כי MobileNet מצליח לדרג בتصורה טובה יותר את סוג הפקים הרלוונטיים בין התוצאות המובילות, ומהווה פתרון יעיל ומכליל יותר בהשוואה למודל הבסיסי. אם זאת חשוב לציין שעדין קיים מקום לשיפור, זאת כנראה בגלל כמות הדadata הקטנה והקשי המשמעותי לחזות באופן מדויק את 2 הסוגים של הפקים.

בחינה של המודל הנבחר על transfer learning mobile net על תמונות מסט האימון:



איור 22: תוצאות של MobileNet בלמידת העברה על סט האימון

בחינה של המודל הנבחר על transfer learning mobile net על תמונות מסט המבחן:



איור 22: תוצאות של MobileNet בלמידת העברה על סט המבחן

בחינה של המודל על תמונות חדשות שהוא מעולם לא ראה, במקורה הזה תמונות של פוקימונים מדור 8 ומדור 9:

הסוגים האמיתיים של הפוקימונים הם: מסווג Sprigatito, Grass Grookey והוא מסווג .Fire Fuecoco.

Grookey (Gen8)
Fairy: 16.6%
Normal: 10.7%
Ghost: 10.1%
Fire: 9.9%
Grass: 9.4%
Water: 9.2%
Rock: 8.3%
Ground: 7.0%
Fighting: 5.5%
Poison: 5.5%
Psychic: 5.4%
Flying: 4.8%
Bug: 4.3%
Electric: 3.3%
Steel: 1.4%
Dragon: 1.2%
Ice: 1.2%
Dark: 1.2%



Sprigatito (Gen9)
Grass: 21.9%
Bug: 15.8%
Psychic: 13.7%
Fairy: 12.8%
Ghost: 5.5%
Water: 4.2%
Rock: 3.2%
Ice: 2.6%
Fire: 2.6%
Flying: 2.4%
Normal: 2.4%
Electric: 2.4%
Fighting: 2.3%
Poison: 2.3%
Ground: 1.1%
Steel: 1.0%
Dark: 0.9%
Dragon: 0.6%



Fuecoco (Gen9)
Fire: 43.3%
Psychic: 17.4%
Ground: 16.2%
Fighting: 11.7%
Rock: 7.9%
Grass: 6.3%
Electric: 5.9%
Flying: 4.9%
Steel: 4.8%
Water: 4.6%
Normal: 4.1%
Ice: 2.9%
Ghost: 2.4%
Bug: 1.8%
Dragon: 1.6%
Poison: 1.4%
Fairy: 0.6%
Dark: 0.6%



פרדיקציות שהמודל הנבחר מבצע על תמונות פוקימונים מדור 8 ודור 9, תמונות שהוא מעולם לא ראה.

סיכום ומסקנות:

פרויקט זה ביצעו שתי משימות סיוג תמונות של פוקימונים **Multiclass Classification** ל-150 פוקימונים מדור 1, ומשימה של **Multi-Label Classification** לחיזי 2 סוגים של פוקימון מדורות 1-7 כאשר לכל פוקimon עד שני סוגים.

שתי המשימות הדגשו את ההבדל בין סיוג "קלאסי" של multiclass לבין סיוג יותר מתקדם וקשה של multilabel, ואת ההשפעה של גודל הדאטה, חוסר איזון בין מחלקות, ורגולריזציה על יכולת ההכללה של כל מודל שבנינו במהלך הפרויקט.

בחלק הראשון עבדנו עם דאטה גדול יחסית (17,000 תמונות בערך), אך עם אתגרים של דמיון חזותי בין מחלקות, שונות גובהה בתוך מחלקה וחוסר איזון בכמות התמונות. ראיינו ש- **Transfer Learning** עדיף משמעותית על מודל הביסליין. המודל הטוב ביותר היה **EfficientNet** Fine-Tuning עם **Top-K Categorical Accuracy** של כמעט 85% כאשר רוב הטעויות שלו היו בין פוקימונים דומים מאוד (למשל התפתחויות אחד של השני).

בחלק השני הדאטה הייתה קטנה (809 תמונות בלבד, לרבות תמונה אחת לכל פוקימון), ולכן אתגרים המרכזים אותנו הם איזון התמונות והחומרה בין המחלקות בין הסוגים השונים של הפוקימונים. ביצענו השוואת CNN baseline בין **MobileNet** ב- Transfer Learning וראיינו שיפור עקבי: מדריך קטגוריאלי (Top-K Categorical Accuracy) הגיע ל-35%, מספר גבוה יותר מהביסליין, אך עדין מוגבל בגלם כמות הדאטה והקיים לחזות שני Labels (2=K) בלבד.

לסיכום, הפרויקט מראה של מידת העברה משפרת ביצועים בשתי המשימות, אך מדגיש את הקושי וההבדל בין משימת ה- Multi-Label שהוא יותר מורכבת משמעותית מביצוע משימה של .Multi-Class.

תוגה'ם:

<https://www.kaggle.com/code/amikhshibu/transfer-learning-tf-inceptionv3-val-cc-82>

<https://towardsdatascience.com/multi-label-classification-of-pokemon-types-with-tensorflow-8217a38038a6/>

<https://www.kaggle.com/code/allenlu112220/pokemon-type-classification>

<https://www.kaggle.com/code/josutk/pokemon-type-identify>

עליזה שף גולדשטיין

סטודנט ב'ן נון