

זיהוי מחלות בפירות

פרויקט בלמידה עמוקה
ורשתות נוירונים

מגיש:

ערן אברהם
302969068

תוכן עניינים

עמוד	נושא
3	מטרת התוכנה
4	קלט
5	פלט
6	תוכנית מימוש
7-9	בחירת המודל
10-11	כלים וספריות
12-13	שיפורים אפשריים
14-15	וידאו וקוד מקור

מטרת התוכנה

▶ מטרת הפרוייקט היא לממש מערכת ראייה ממוחשבת לצורך זיהוי פגמים בפירות-הדר.

▶ המערכת תבצע זיהוי של פגמים המשותפים לפירות-הדר שונים.

▶ המערכת תקטלג תמונות של פירות-הדר על פי סוג המחלה שתזהה בהן.

קלט

- ▶ המערכת מקבלת כקלט אוסף קבצי תמונות (או קובץ תמונה בודד), המציגות פרי הדר.
- ▶ התמונות יכולות להיות בפורמט RGB.

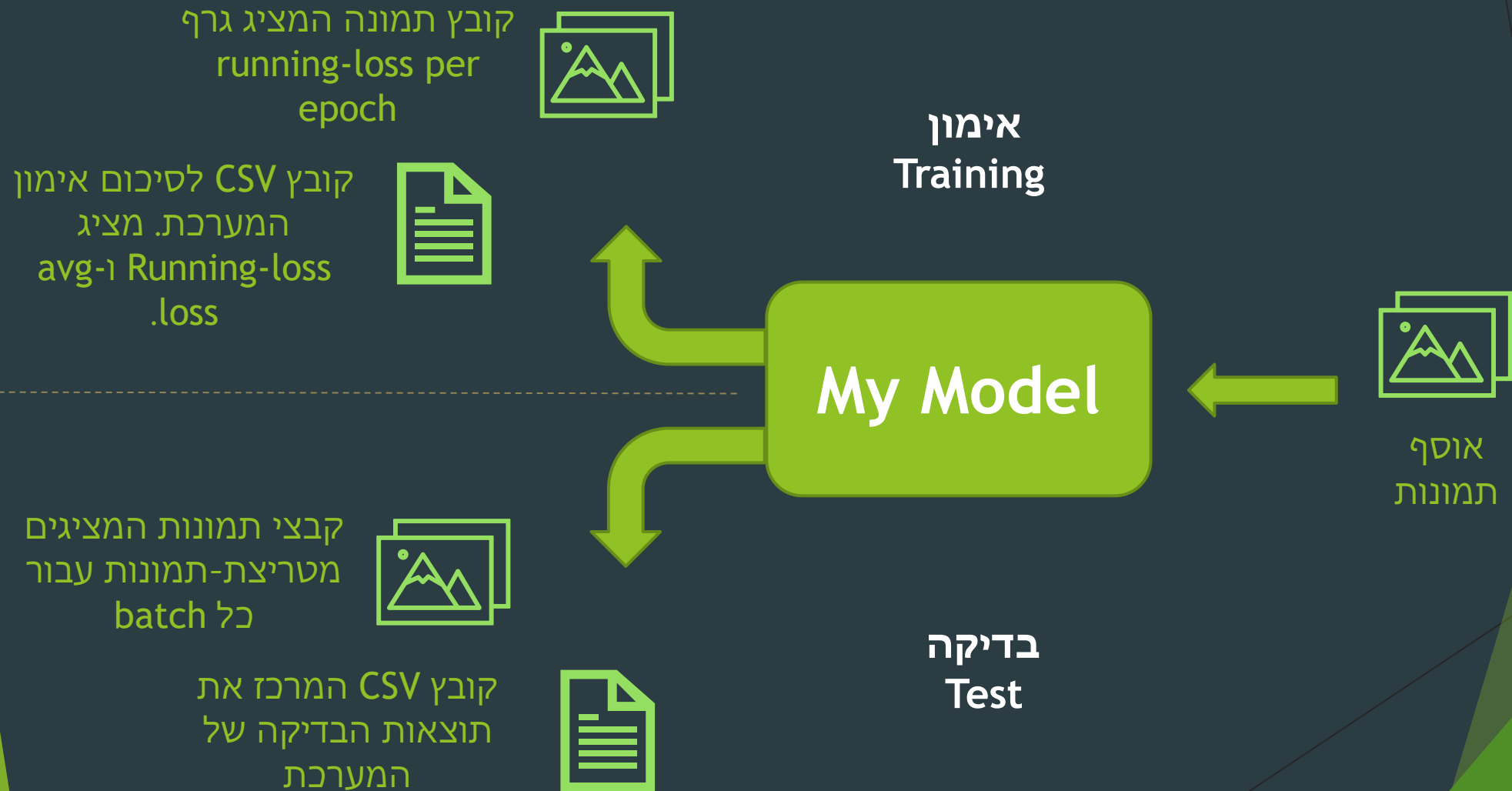
פלט

- ▶ המערכת תוציא פלט בשני פורמטים: CSV וקבצי תמונות.
- ▶ המערכת תיצור קובץ CSV המפרט את תהליך בדיקת-המערכת. בנוסף, המערכת תיצור קובץ תמונה לכל batch עם מטריצת תמונות ותחזיות מתאימות.
- ▶ המערכת תדע לזהות אחת מבין 2 המחלות הנתונות:

• Black Spots

• Canker

תוכנית מימוש



בחירת הדאטה

- ▶ הדאטה-סט שנבחר, נלקח מאתר **Kaggle.com**, ומכיל תמונות עבור שתי מחלות שונות של מחלות פירות-הדר.
- ▶ לאחר צמצום כל התמונות אשר אינן מתאימות לאימון המערכת (רקע לא טוב), נשאר דאטה-סט מצומצם.
- ▶ הדאטה סט כולל 334 תמונות בכל קטגוריה, מספר קטן מידי אך ישמש כ"ממלא מקום" בזמן בניית המערכת.

בחירת המודל

- ▶ מודל רשת ראשון שנבחר, נלקח מפרוייקט לזיהוי צמחים, אך תהליך הלמידה נתקע במינימום מקומי ולכן הוחלט לנסות מודל אחר.
- ▶ **לכן**- מודל הרשת שנבחר הוא ResNet50 מתוך ספריית Pytorch.
- ▶ הרשת מתאימה למשימת Classification, ולא נעשה שימוש במשקלים מאומנים מראש (מתוך ספריית Pytorch).
- ▶ מכיוון שרשת resnet50 מחזירה וקטור פלט של 1000 מחלקות שונות, (בהתאם ל-ImageNet dataset) חיברנו את שכבת הפלט של הרשת באופן Fully-Connected לשכבת output נוספת המכילה 2 נוירונים.
- ▶ כך, המודל שלנו עושה שימוש ב-resnet50 לבחירת 2 מחלקות שונות שהגדרנו בדאטה.

היפר פרמטרים

- ▶ מכיוון שהדאטה שעליו עבדנו היה מצומצם, לא בוצעו יותר מ-5 epochs לאימון המערכת.
 - ▶ המטרה היתה לבצע חזרות רבות עם batch-ים קטנים, כך שבאימון המערכת יתבצעו תיקונים רבים אך מתונים!
 - ▶ תיקונים בוצעו במקרים של **overfitting** (אימון גס מידי) או **underfitting** (עצירה במינימום מקומי).
 - ▶ הפרמטרים שנבחרו הם:
- EPOCHS = 5
 - Batch Size = 16
 - Learning Rate = 0.0001

כלים וספריות

▶ ספריות קוד בהן נעשה שימוש:

- Pytorch - ספרייה עם כלים לבניית רשתות נוירונים וכן כלים לניהול ועבודה נוחה עם הדאטה שלנו.
- Pytorch.TorchVision - ספרייה זו מביאה לנו מודלים מוכנים של ANN וכן כלים נוספים לעבודה איתם.
- Numpy - ספרייה לחישובים מתמטיים ועבודה עם וקטורים.
- Matplotlib - אוסף כלים להצגה ויזואלית נוחה של המידע שלנו, כמו גרפים.
- Pandas - בזכות ספרייה זו, יכולנו לעבוד בקלות עם מידע טבלאי ולשמור אותו בקלות לתיעוד להמשך.

כלים וספריות

▶ נלקח ממקורות חיצוניים:

- מודל הרשת: מודל רשת הנוירונים שבו נעשה שימוש - ResNet50 - נלקח מספריית הקוד Pytorch.

▶ פיתוחים מקוריים:

- מודל סופי - מודל הרשת הסופי עוטף את רשת resnet50 ומנתב אותה למספר מחלקות כרצונינו.
- ארגון הפלט - אלגוריתמים לניהול פלט המערכת -באימון ובבדיקה- ושמירתו בפורמט שקל לצפות בו (תמונות וגרפים) ובאופן שקל לעשות בו שימוש בהמשך (שמירת מידע טבלאי בCSV).
- פונקציות אימון ובדיקה - תהליך האימון והבדיקה של המערכת פותח באופן עצמאי, על פי הדוקומנטציה של ספריות הקוד, מאמרים אינטרנטיים וניסוי-וטעייה.

שיפורים אפשריים

▶ שיפורים אפשריים במערכת הקיימת:

- דאטה - ראשית, וכפי שצויין קודם לכן, הדאטה שנעשה בו שימוש לצורך הפיתוח היה מצומצם מידי, ולא מספיק מגוון. על מנת שהמערכת תוכל לבצע תהליך למידה רחב יותר המתאים לעולם האמיתי, עלינו ליצור דאטה-סט מקיף הרבה יותר. אוסף תמונות גדול יותר. פירות הדר נוספים עם אותן מחלות.
- מודל - מודל הרשת שבה נעשה שימוש הוא מודל כללי ל-Classification. ניתן לבחון מודלים מתחרים, או לחשוב על מודל חלופי משלנו לארכיטקטורת רשת הנויירונים.

שיפורים אפשריים

שיפורים להמשך פיתוח עתידי של המערכת: ►

- **Transfer Learning** - רעיון מרכזי בפיתוח להמשך, הוא ביצוע "העברת-למידה" על מנת לאמן את המערכת שלנו להכיר סוגי מחלות חדשות. אנו נבצע שימוש במודל המאומן שלנו ונאמן אותו עם דאטה חדש בכדי להוסיף לא מחלקות זיהוי חדשות.
- **Object Detection** - רעיון נוסף שנדחה משלב הפיתוח הראשוני, הוא ניצול טכנולוגיה מתחום "ראייה ממוחשבת" לשם הוספת אספקט של זיהוי אובייקטים בתמונה. באופן זה המערכת שלנו תוכל לבצע "לכידה" של איזורי מחלה נפרדים באותו הפרי ולקטלג אותם בנפרד.
- **UI** - לשם נוחות השימוש, והפיכת המערכת שפיתחנו לכלי זמין, עלינו לפתח ממשק משתמש נוח להפעלה במחשב או מכשיר נייד. כרגע, המערכת עדיין מופעלת מופעלת בסביבת פיתוח תוכנה (IDEA) ומציגה תוצאות באמצעות ספריית Matplotlib.

ביבליוגרפיה

מקורות מידע: ►

- <https://blog.jovian.ai/using-pytorch-to-classify-flowers-ce297310a412/>
- <https://www.nature.com/articles/s41598-021-96103-2/>
- <https://efficientdl.com/faster-deep-learning-in-pytorch-a-guide/>

וידאו וקוד מקור



קוד תוכנה: ►

• [קישור לקוד המקור באתר GitHub](#)

הצגת המערכת - וידאו: ►

• [הסבר על הקוד](#)

• [הצגה מוקלטת של המצגת](#)

• [הרצת התוכנית](#)

