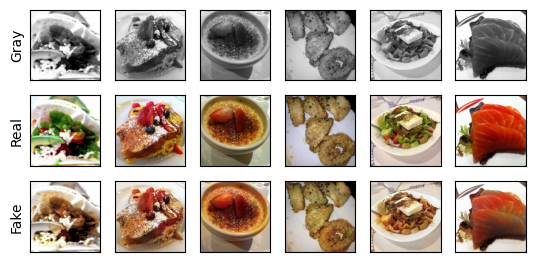
*פרויקט סופי בלמידה עמוקה – צביעת תמונות*

**אלברט כיאט – 212747026, דוד אפיק – 318267721**

# תיאור הבעיה

עלינו ליצור מודל שמקבל תמונות בשחור-לבן וצובע אותן.

דוגמה לפלט של המודל הסופי מה-testset:



## דאטה

בחרנו ב-database "[Food-101](https://data.vision.ee.ethz.ch/cvl/datasets_extra/food-101/)" שבו יש 101,000 תמונות של אוכל, אבל לאימון לקחנו 8,000 תמונות באקראי (ועוד 2000 בשביל ה-testset).

פורמט התמונות:

בחרנו להמיר את התמונות למרחב LAB מכמה סיבות.

במרחב LAB, הערוץ L הוא פונקציה (לא לינארית) של התמונה בשחור לבן, והערוצים AB הם ערוצי הצבע. במרחב הזה אנחנו יכולים להפריד את התמונה הצבועה לערוץ L, ה-label, ולערוצים AB, ה-feature, ובכך להפוך את הבעיה להיות: בהינתן תמונה באפור (הערוץ L), המודל צריך לייצר את הצבע (הערוצים AB).

סיבה נוספת לשימוש במרחב הזה היא שמרחק בין תמונות מייצגת את ההבדל בנראות של הצבעים יותר מאשר המרחב RGB.

# ארכיטקטורה

### הרשת cGAN

הארכיטקטורה שמימשנו היא Conditional GAN (cGAN). בשונה מ-GAN רגיל, ב-cGAN משתמשים גם ב-label כדי לייצר דאטא (וזה ה-condition).

cGAN מתאים לבעיה שלנו כי ה-generator מקבל תמונת אפור שהוא צריך לצבוע, ולא מייצר תמונה חדשה מרעש כמו ב-GAN. בנוסף, ה-discriminator מקבל את ה-feature ואת ה-label כדי שהוא יסווג את התמונה המלאה, ולא רק את הצבע.

פונקציית ה-loss ב-GAN היא:

כאשר מתפלג מתוך ה-data האמיתי שמייצרים, ו- הוא רעש שמתפלג בצורה כלשהיא.

פונקציית ה-loss שלנו (cGAN) היא:

כאשר הוא התמונה בשחור לבן, הוא הצבע של התמונה ו- רעש.

*בנוסף ל-cGAN רגיל, הוספנו ל-loss את פונקציית ה-MAE (mean absolute error), או , בין הצבעים שה-generator מייצר לבין הצבע האמיתי, והוא מוכפל במשקל . עם ההוספה הזאת, כשעושים צעד בגרדיאנט של ה-generator, הוא מנסה לא רק לעבוד על ה-discriminator, אבל גם להתקרב לצבע האמיתי עם משקל .*

*לבסוף, ה-objective הוא:*

סך הכל, ברשת יש 6,192,451 פרמטרים (2,270,754 ל-Unet ו-3,921,697 ב-discriminator).

זמן האימון לרשת היה סך הכל 5 וחצי שעות, שעה וחצי ל-pre-train (15 epochs אימון רק על ה-Unet) ו-4 שעות לאימון (40 epochs).

#### GENERATOR – UNET

השתמשנו ברשת UNET דומה לזאת שראינו בקורס.

להלן תרשים הרשת:

מקרא מודולי הרשת:

A screenshot of a computer

Description automatically generated

רשת ה-UNET

A diagram of a computer

Description automatically generated

הרשת מקבלת תמונה עם ערוץ אחד (L – רמות האפור של התמונה), מוסיפה לה רעש ומחזירה את שני הערוצים של הצבע (A ו-B).

#### DISCRIMINATOR

השתמשנו ברשת fully-convolutional בעלת 6 שכבות:

A group of rectangular labels

Description automatically generated

הרשת מקבלת תמונה בעלת 3 ערוצים L, A, B, ומחזירה ערך בין 1 ל-0 שאומר אם התמונה אמיתית (1) או מזויפת (0).

# אימון

לפני האימון של ה-cGAN, אימנו רק את ה-Unet לקצת כדי שהוא יתחיל במצב יותר טוב. ה-Loss שחישבנו ב-pretrain הוא (או MAE).

את ה-Loss של ה-cGAN חישבנו עם ה-class GANLoss, שמקבל חיזוי של ה-discriminator, בנוסף ל-bool שאומר אם ה-data אמיתי או לא.

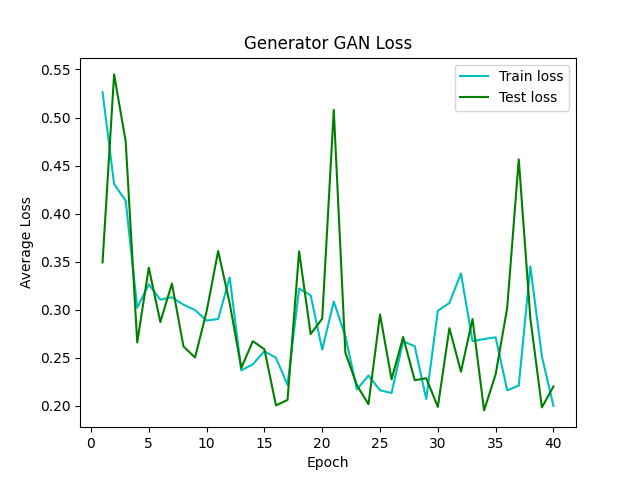
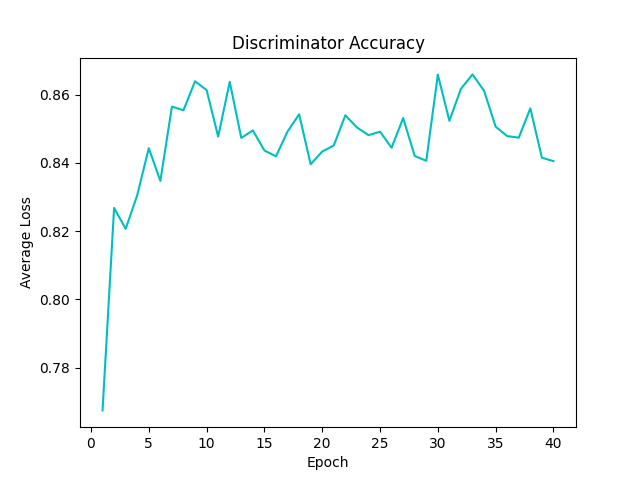
המחלקה משתמשת ב- Torch.nn.BCELoss שמחשב:

אם ה-data אמיתי, הביטוי שמחושב הוא , ואם ה-data מזויף, הביטוי שמחושב הוא .

באימון ה-discriminator חוזה עבור data אמיתי, ו- עבור data מזויף שה-generator יוצר, וה-Loss של ה-discriminator שמחושב הוא:

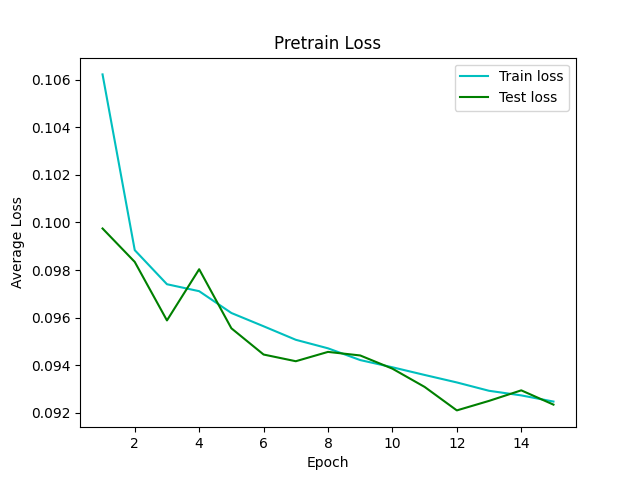
*ה-generator מחשב*

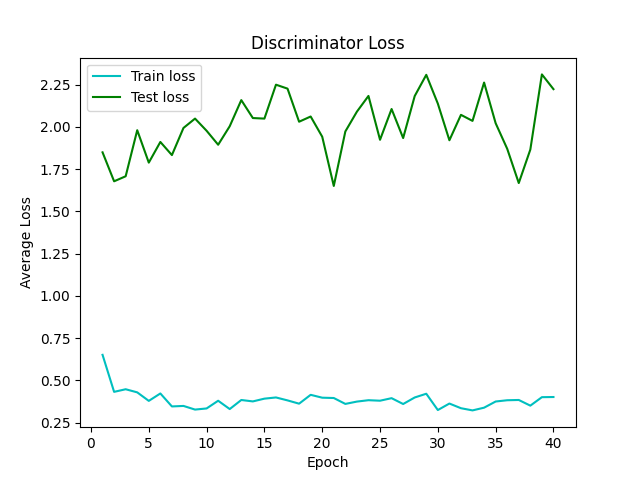
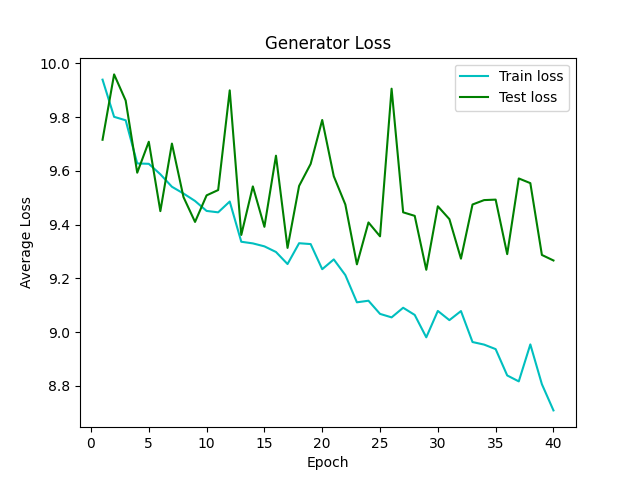
באימון של ה-cGAN, כדי לאזן את הכוחות של ה-discriminator וה-generator, עבור כל batch (בגודל 16) העברנו את התמונות האמיתיות והמזויפות ב-discriminator, ואם הוא צדק ברמה מספיק גבוהה המודל מדלג על האופטימיזציה של הפרמטרים שלו כדי לתת סיכוי ל-generator ללמוד לרמות אותו.

את האפקט אפשר לראות בגרפים הבאים כאשר ה-discriminator לא עולה על accuracy גבוה מידי:

# תוצאות

אימן ה-pretrain, כשה-Loss הוא MAE בין הצבע האמיתי והמזויף:



אימון המודל:

תמונות מה-testset:

A collage of different foods

Description automatically generated

A collage of different foods

Description automatically generated



# הרצה

כדי לצבוע תמונות, יש לשים את התמונות בתיקייה “paint” ולהריץ את “paint.py”. לאחר מכן יופיעו התמונות הצבועות באותו תיקייה.

כדי להריץ את האימון, צריך להוריד את ה-dataset מהאתר "[Food-101](https://data.vision.ee.ethz.ch/cvl/datasets_extra/food-101/)" ולשים את התיקייה  
“food-101” לתיקייה “data” ולהריץ את “data\get\_subset.py”.