סדנא מעשית בלמידה עמוקה

מטלת מחקר 1

CiFAR10

מגישים:

עידן רזי, 204475255

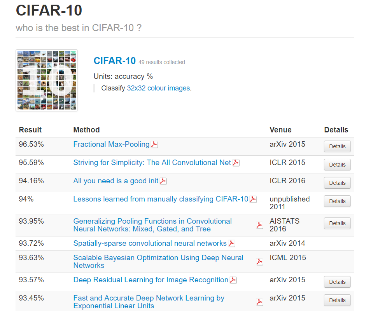
אסף ולודבר, 204233936

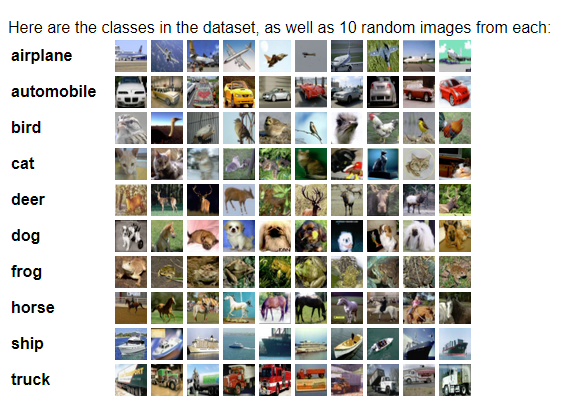
1. פרטי המאגר:
   1. גודל מאגר המידע : 50000 תמונות אימון ו10000 תמונות מבחן, בגודל 32\*32 פיקסלים צבועים (32,32,3)
   2. כל תמונה בגודל 32\*32 ומכילה 3 ערוצי צבעים בגודל 0-255 .

מאגר המידע מכיל 10 מחלקות שונות(סעיף e)

מאגר המידע אינו נדרש עיבוד קודם מכיוון שקיימת התפלגות אחידה בין המחלקות במאגר , בנוסף התמונות אינן שייכות למספר מחלקות, כלומר כל תמונה מסווגת למחלקה אחת בלבד.

ניתן להשתמש באוגמנטציה אופקית, כלומר הסטה של התמונות ימינה או שמאלה ובנוסף היפוך מראה.

* 1. מאגר המידע מאוזן כפי שציינו מקודם, כל מחלקה במאגר האימון מכילה 5000 דגימות (סה"כ  50000).
  2. ניתן לראות כי קיימים מודלים שונים אשר מהווים benchmark למאגר הנתונים בו אנו משתמשים:



1. על פי הגדרות המאגר Cifar 10 מאגר האימון מכיל 50,000 דוגמאות במאגר האימון כאשר הם מפוזרים רנדומלים במאגר ובצורה אחידה. לכן, על ידי שימוש בvalidation split בעצם עשינו חלוקה כזו שלקחנו את 20 אחוז מהמאגר לvalidation והוא יכיל אוסף רנדומלי מכל הסיווגים האפשריים שצורה השואפת להיות אחידה.
2. במהלך הניתוח שביצענו על הנתונים שמנו לב שרשת פשוטה על המאגר הנוכחי לא מספיקה על מנת להביא לתוצאות גבוהות, קיימים סוגים שונים של סיווגים שהמאפיינים שלהם דומים מאוד וקיימים מאפיינים שהחשיבות שלהם לוקחת משקל גבוה מאשר אחרים. על כן ככל שהתקדמנו במחקר הוספנו פילטרים בכמות שלא תעשה over fitting אבל תביא בחשבון את כמות המאפיינים הנדרשת לרמת דיוק גבוהה וביצענו נרמול של הנתונים ועיבוד מחדש על מנת להגדיל ולשבח את המאגר הקיים למען זיהוי מדויק יותר והפרדה בין הסיווגים בעלי המאפיינים הדומים.  
   להלן דוגמאות המשקפות את ההתפתחות של המודל לאורך הזמן עד הגעה למצב בו האימון והמבחן כמעט באותו קו וברמת זיהוי גבוהה מאוד, עד 80%.

מודל 1: במודל זה השתמשנו ברשת פשוטה עם 4 שכבות נסתרות בלבד ללא כל עיבוד של הנתונים.

ניתן לראות שלא רק שאחוז הסיווג נמוך יחסית כיוון שמאפיינים רבים דומים בין סיווגים שונים גורמים לטעויות רבות, גם גרף המבחן לא אחיד ולא קרוב לרמת הדיוק של האימון.

|  |  |
| --- | --- |
|  | Model - 63.33% Accuracy |
|  |

מודל 2: על מנת לשפר את דיוק המבחן הוספנו שתי שכבות פילטור נוספות על מנת להוציא מאפיינים נוספים שאולי ישפרו את התוצאות. ניתן לראות בבירור ששכבה נוספת העלתה את גרף המבחן וייצבה אותו כך שנלקחו לידי ביטוי מאפיינים רבים שפוספסו לפני כן. עם זאת, עדיין קיים דיוק נמוך וזאת בעקבות דמיון מאפיינים בין הסיווגים השונים.

|  |  |
| --- | --- |
| Model - 68.1% Accuracy |  |
|  |

1. לאחר בחינה של הפלטים מהמודלים השונים שניסינו הבחנו כי קיימות טעויות בזיהוי שנפוצות אצל כולם.  
   המודלים לא מצליחים להפריד ברמת דיוק גבוהה מספיק תמונות בהן מופיעים מטוסים, ציפורים וספינות. מבחינה של התמונות אנו סוברים כי הסיבה לשגיאה היא הרקע של השמיים שחוזר בכל שלושת הסיווגים וכן כי ציפורים ומטוסים לעיתים בעלי צורה דומה.

רכבים ומשאיות גם כן לא מופרדים בצורה מדויקת מספיק, הרבה מכלי הרכב מסווגים כמשאיות אם כי רוב התמונות של המשאיות כן סווגו נכונה.  
כאשר מדובר בחיות, הרבה מהחיות סווגו כצפרדעים, ככל הנראה בגלל הרקעים הדומים של הטבע. עם כי צפרדעים סווגו כמעט בצורה מושלמת יחסית בכל התמונות. כמו כן הסוסים והאיילים, הכלבים והחתולים גם כן בשל צורתם הדומה סווגו שלא נכונה.

שלוש דרכים אפשריות לדייק ולתקן את הסיווגים הללו הן:

1. לנרמל את הערכים בשכבות הנסתרות, כך נוריד את החשיבות של פיצ'רים מאחרים וכך אולי יהיה ניתן להפריד בין סיווגים שונים, לדוגמא ציפורים ומטוסים שקיימים להם מאפיינים רבים שדורשים התייחסות על מנת להפרידם.
2. הוספת padding לגבולות התמונה וכך לתת יותר התייחסות ממה שניתן קודם לרקעים ולקצוות התמונה השונים.
3. לבצע אוגמנטציה כזו שתוסיף עוד תמונות לאימון, כמו לדוגמא להפוך אפקית את התמונה, כיוון שהתמונות שלנו מכילות עצמים שניתן להשתמש גם בתמונות המראה שלהם, אוגמנטציה מסוג זה יכולה להוסיף לדיוק טוב יותר.
4. אנו בחרנו להוסיף שכבות נרמול ולהוסיף padding, דברים שחשובים בראש ובראשונה בלמידה נכונה ולהלן התוצאות:

מודל 3: על מנת לתת יותר התייחסות למאפיינים חלשים יותר הוספנו שכבת נרמול. כעת ניתן לראות שיפור משמעותי בדיוק אימון לצד דיוק מבחן שעדיין לא מצליח להשתפר כפי שהיה רצוי ובעל שגיאת מבחן גבוהה משגיאת האימון.

|  |  |
| --- | --- |
| Model - 73.1% Accuracy |  |
|  |

מודל 4: במודל זה ניסינו להוסיף padding במחשבה שהתייחסות לקצוות התמונה ייתנו זיהוי יותר מהימן אך נראה כי התוצאות לא השתנו.

|  |  |
| --- | --- |
| Model - 73.35% Accuracy |  |
|  |

מודל 5: לאחר שהבנו כי הנרמול העלה את התוצאות ניסינו להוסיף שכבה נוספת שלו לראות עד כמה הוא יכול לשפר את המודל שכבר נורמל בשכבה קודמת. ניתן לראות כי נרמול שכבה נוספת הוסיף לשיפור הביצועיים של המודל.

|  |  |
| --- | --- |
| Model - 75.42% Accuracy |  |
|  |

מודל 6:

כעת הבנו שמיצינו את האפשרויות של הנרמול ולכן ניסינו לשחק עוד עם המידע שקיבלנו על מנת להוסיף דוגמאות ללמידה ולשפר את הסיווגים על ידי כך. ביצע אוגמנטציה של הנתונים כך שתמונות אשר האובייקט מופיע בהן קטן במרכז התמונה ואחרות בהן האובייקט מכיל את כל התמונה יקבלו ביטוי דומה ולכן עשינו זום של עד 20% לתמונה לצד נרמול הנתונים בין 0 ל-1 וסיבוב אופקי של כל תמונה, מההבנה שכל אובייקט שלרשותנו ניתן לראות כתמונת מראה ולזהות אותו באותו סווג.

ניתן לראות כי גרף המבחן עולה במידה רבה ומתחיל להתקרב לגרף האימון מה שגם הביא לשגיאה נמוכה הרבה יותר.

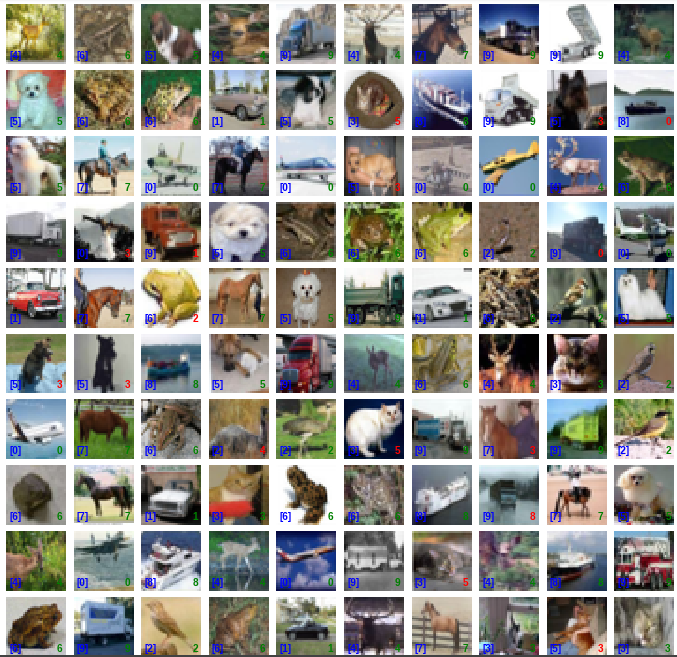
|  |  |
| --- | --- |
| Model – 76% Accuracy |  |
|  |

מודל 7:

ניסינו לבחון על רמת ההשפעה של הזום על המודל ולכן ביצענו שוב את אותו אוגמנטציה רק עם זום של 50%.

אכן ניכר שיפור משמעותי ולכן ניתן להבין כי הזום מאפשר ליצור דוגמאות נוספות בעלות משמעות לזיהוי בין סיווגים. ניתן לראות שגרף הדיוק והשגיאה כמעט זהה בין האימון למבחן.

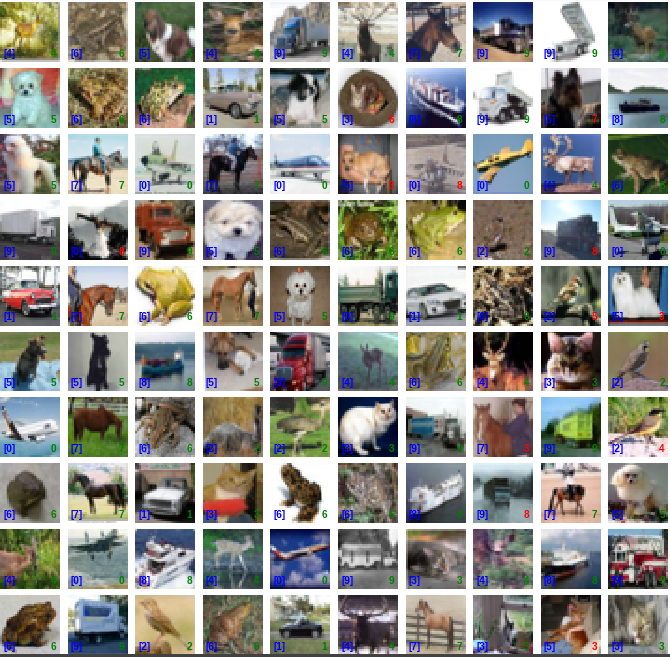
|  |  |
| --- | --- |
| Model – 78.09% Accuracy |  |
|  |

להלן דוגמא לזיהויים שונים עם המודל:

מודל 8:

אם במודל הקודם ראינו את השפעת הזום על הדוגמאות, רצינו לראות אם שינוי נוסף יטיב עימנו. כאשר העלנו את הזום מעל 50% הייתה הרעה של התוצאות אך לאחר שהורדנו אותו ל30 אחוז דווקא הדיוק היה הרבה יותר טוב.

|  |  |
| --- | --- |
| Model – 79.16% Accuracy |  |
|  |

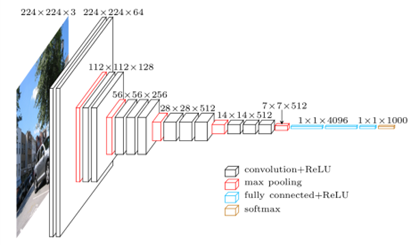
להלן דוגמא לזיהויים שונים עם המודל:

1. רשת טרנספר

החלטנו להשתמש ברשת vgg16 מכיוון שזו רשת שיחסית דומה במבנה שלה לרשת שבנינו בעצמנו בסעיף הקודם כדי שנוכל לבצע השוואת ביצועים. השתמשנו ברשת הvgg16 מאומנת במשקלי 'imagenet' ללא השכבות האחרונות, ועליהם הוספנו שכבות נוספות.

כחלק מהכנת המאגר, לאחר בחינה של הרשת הקיימת, ביצענו נרמול נתונים לטווח של 0-1 בעזרת חילוק הנתונים בערך המקסימלי 255.

מבנה הרשת 16VGG:



|  |  |
| --- | --- |
| מבנה הרשת | דיוק 0.36% |
| https://lh6.googleusercontent.com/vyPNsU7I6e-dQ48qUrv0qyDZxUWEYUsW9c5f9SlA8Z1TR6qrmjBLkvyQ-cFA3DsnD3XZ3-iF_oHp1VkfoWdtepGmoyiR5ws2goYbhu2vvgpownM7HN37xO6NDLgOrjAlb498QqAw | https://lh6.googleusercontent.com/EacLlzldgVJ0T0mCDZ3V49Bu7tulPLgnVyNnNJowVn2t9HdHUv6dkdCLYNBXelrB7AFUJcpqFyc71i7Z_s0SEodyWV2wF88ya0RIb-YAu87F32YR9yq1bIflNnPmI6dwYQLf-5tP |

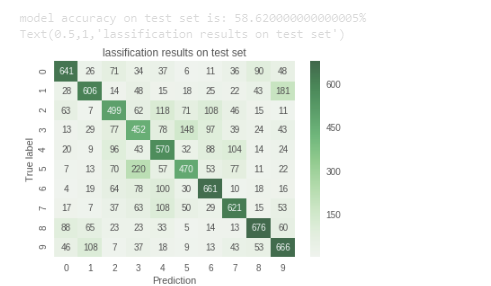
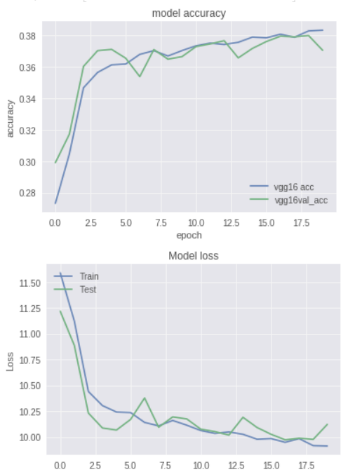
**מודל ראשון לרשת המאומנת:**

ניתן לראות אחוז דיוק מאוד נמוך של 0.36, במיוחד ביחס לרשת מהסעיף הקודם. לכן, היינו צריכים לבדוק מה לא עשינו כשורה.

מודל 2:

כדי לשפר את אחזור הדיוק השתמשנו ב בשכבת Globalaveragepooling , במקום Flatten

ניתן לראות ששכבת הglobal צמצמה את מבנה הרשת לוקטור אחד בגדול 512.

https://lh3.googleusercontent.com/0JvIFLAVhSysfSNEs-xTYhaXOdszWKpXdcX08F5kSzxTTJtLk-mwCJq2ZUSGeelfMvMQuP95HOfXrWHEMejpyVx9oYFhYbm_9PM6wGe2hQKDmalw9MzMMPdE1515M5omQyJ-b7da

תוצאות המודל השתפרו אך עדיין היו נמוכות מאוד, 0.58, וניתן לראות ע"פ גרף הדיוק, שממשיך לעלות, כי אימון ארוך יותר כנראה היה משפר את דיוק המודל.

התוצאה הפתיעה אותנו מכיוון שמדובר ברשת מאומנת וצפינו לראות אחוזי דיוק גבוהים.

**החשד העיקרי** שלנו הוא חוסר התאמת רשת הtransfer לגודל הinput שלנו (32\*32) לעומת (224\*224).

חיפוש באינטרנט בdocumentation keras הראה שניתן להזין גדלי תמונות שונים (מינימום 32 כמו אצלנו במאגר), ובנוסף שימוש בפונקציה preprocess\_input שמגיעה עם vgg16 יכולה לשפר את הביצועים.

מודל 3:

ביצוע עיבוד מקדים לתמונות בעזרת הפונקציה preprocess\_input שמגיעה עם keras אכן שיפר את ביצועי הרשת ל64 אחוז דיוק אימון.

|  |
| --- |
| דיוק: 64% |
| https://lh5.googleusercontent.com/t4-g84hpWcnvjHDVqpZritRH3hpP1kTYpaZjO5Vpkkr5nQhYuRtHj9bbkzmmv9kj_aUQ628dMNmj7rY7bqqJyvvdz2O4kKWOvwgBH_C4_B4WiH-iXBYaM79N5MRYfTeBHc58PDdc |

אבל ע"פ הגרפים ניתן לראות הבדל משמעותי בין דיוק האימון לדיוק המבחן.

בנוסף, ע"פ גרף פונקציית ההפסד ניתן היה לראות overfitting.

**\*ניסיון של הוספת רשת fully connected layer לפני שכבת הsoftmax הורידה את אחוזי הדיוק.**

לאחר בחינה נוספת באינטרנט ומציאת המאמר הבא: Network In Network

<https://arxiv.org/pdf/1312.4400.pdf>

**הגענו לכמה מסקנות:**

* שימוש בglobalavg/max בעצם מבטל את הצורך בשכבת FC ויש לחבר אותו ישר לשכבת softmax
* שכבת FC יכולה לגרום overfitting במיוחד כאשר אינה מאומנת (כמו התוצאות של הגרף הקודם שהצגנו).
* ניתן לראות את שכבת  globalavg/max כמין רגולטור (בדומה למה שלמדנו בשיעור) ובעצם מבליט את התכונות שנלמדו עד כה ולכן מהווה שכבה טובה לפני שכבת הסיווג.

לכן החלטנו לבטל את שכבות ה FC כיוון שהבנו שהדבר לא הגיוני, כלומר, לחבר את שכבת הglobal 512 ל שכבת FC בעלת 1024 קודקודים.

מודל 4:

ניתן לראות ע"פ הגרפים שאין overfitting, אבל אחוז הדיוק במהלך האימון יותר נמוך.

|  |  |
| --- | --- |
| מבנה הרשת | דיוק : 58% |
| https://lh6.googleusercontent.com/rcHsXySnpAJ3CezghiHbhf0VJDQ0lgQQ3CjuOSX_LcYC7z4o8DRvFY4AMtLRnqbnEse5wI2l1ndVnvwzQDg04e4Kyvymup9jsVDyMAMOnG6ZvS6bvi6Dr7nbkcdkidAPAKmZjQv3 | https://lh4.googleusercontent.com/635LyTz32-bJGZ3I_kPaAZJx_-AGvWYg1U5b7OE_ji-SByFSw4ntmdEGwO60MyH36SjoiSsdt6e6nDPI1Nk120vXEJ9XUUsT-Gy-80YaoaV_3Q1TaOKev6lT28_EKGhlxC8qrhx0 |
|  |

**מודל 5** : ביצענו חיפוש  נוסף באינטרנט של מספר דוגמאות לשימוש ברשתות transfer של VGG, וראינו שימוש במודל משכבת maxpool - block3 של רשת הVGG .

לאחר מספר ניסיונות הוספנו שכבת נרמול לפני חיבור השכבות החדשות שלנו.

בנוסף שכבת FC של 256 נוירונים, שכבת dropout ושכבת הסיווג האחרונה SOFTMAX.

הצלחנו לבצע שיפור משמעותי ולהגיע **ל85% דיוק. אולם, עדיין לפי גרף הדיוק והאיבוד ניתן לראות הפרשים גבוהים בין האימון למבחן והגענו למיצוי של הרשת מוקדם יחסית וניתן היה להפסיקה באמצע.**

|  |  |
| --- | --- |
| מבנה הרשת | דיוק : 85% |
| https://lh3.googleusercontent.com/vhu4aTGlwshPNlJ6m5GlV4K01Ph8ELxsEnJlMLWQgjH8mm5Ma6VNtxveACvmTwqaNHDC4TcVyp662elfMm53X0vDGNG4VnFqZvT7LySOUnodjhE88LVwoDwBF6BeF7L0kKw46P2m | https://lh4.googleusercontent.com/V6VmfEJoSV8PWjVRnMMZPGWKfYMJt4dKhiHXK-L1alS5ndoj68CKpjhSsYcnophVICEyz8PP523goeo66hBsCtlNxRUEElaon4JXsX-UGDmv7mt6MdVXYrH4YEU1-16VE5mD9Z2Rhttps://lh3.googleusercontent.com/OsytfMbw5C3FXHhtok79erU9JJLNiVehRM0FleST3t0kbassNh1mt23qJJZdAu-4vbU_P0KBtWdHB9rF2nA6GAxwwFMkyK0klaLU0jK6ZzqV3xP4lE3N-4hB0mZ2rzjtXJvHJNqc |

**מודל 6:**

ניסינו לשפר את המודל באמצעות אוגמנטציה אבל לאחר מספר ניסיונות הצלחנו להגיע לשיפור די קטן של 87% דיוק. בנוסף, ניתן לראות שגרף האימון והמבחן, הן בדיוק והן בהפסד, כמעט זהים מה שמראה על למידה נכונה.

|  |  |
| --- | --- |
|  |  |
|  |  |

כמה דוגמאות לתמונות שזוהו ע"י המודל.



**מסקנות לחלק C:**

לאחר ניסוי עם ורסיות שונות של שימוש ברשת מאומנת VGG – הדיוק המרבי שהשגנו הוא באמצעות

הסרת שכבה נוספות מהרשת למעט 4 השכבות האחרונות ואימון שכבת FC משלנו לטובת המאגר CIFAR10.

הדבר נובע ככל הנראה מגודל התמונות במאגר (32\*32) ולעומת תמונות שהרשת שVGG16 קיבלה ב –

Imagenet(224\*224).

בנוסף נרמול הנתונים בשלב הpre-processing ובמהלך האימון בעזרת שכבת נרמול עזרה להגיע לתוצאות

טובות יותר.

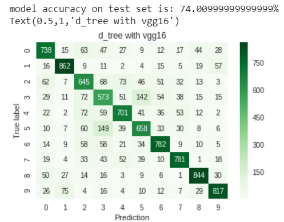
**D**.   
בחלק האחרון בחרנו להשתמש במודל שאימנו מסעיף C כדי לחלץ פיצ'רים ובמודל עץ החלטה על

מנת לסווג את התמונות.

בתור הכנה למודל העץ הסרנו את 2 השכבות האחרונות של המודל שלנו (softmax , dropout).

ביצוע predict על קבוצת האימון שלנו על מנת לקבל טבלת פיצ'רים בגודל (50000,256).

ושימוש במודל עץ פשוט לביצוע חיזוי.



הגענו לתוצאה של 74% דיוק - פחות טובה מאשר פונקציית הSOFTMAX.

ככל הנראה בגלל כמות הפיצ'רים הגדולה (256) שיכולה להוביל לoverfitting.

ככל הנראה חילוץ פיצ'רים מיטביים מהטבלה שהמודל שלנו ייצר באמצעותchi square ובנוסף ביצוע דסקרטיזציה לקבוצות בכל פיצ'ר היה מיטיב עם המודל.

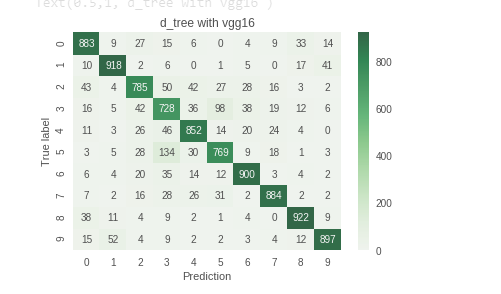
מודל נוסף שניסנו הוא SVC .

הגענו לתוצאה טובה מאוד בעזרת השילוב של המודל שלנו וSVC. 85.38% דיוק.

המודל מתאים יותר לטבלת הפיצ'רים מכיוון שהפיצ'רים רציפים וגם מודל הSVC יודע להתמודד עם כמות רבה של פיצ'רים כיוון שהוא יודע לוותר עליהם במקרה והם לא וקטורים תומכים.

באופן דומה יצרנו טבלת פיצ'רים מהמודל שלנו בגודל (50000,256) .

אימנו מודל SVC , ובחנו את קבוצת הtest דרך יצירת טבלת נתונים בעזרת המודל שלנו (10000,256) וחיזוי בעזרת הSVC.



**ניסוי של רשת הטרנספר עם תמונות חדשות**

ניתן לראות שהמודל לא הצליח לזהות תמונה של הכלב של עידן ושל הצפרדע אבל הצליח לזהות שהדמות המצוירת "בוג'ק" היא סוס.

