דו"ח פרויקט למידת מכונה

מערך הנתונים:

עבור פרויקט זה, בחרנו את מערך הנתונים CIFAR ,100-המורכב מ-000,60 תמונות RGB 32x32 הכולל 100 מחלקות .מתוך 000,60 התמונות, מערך הנתונים מחולק ל-000,50 דגימות אימונים ו-000,10 בדיקות .עבור הפרויקט שלנו, נשמיט 10% מהמחלקות ונשתמש בהם כמחלקה אחת עבור בחינת ההתנהגות של שיטת OOD ונלמד איך האלגוריתמים מתמודדים עם המחלקה החדשה.

שאלות מחקריות שנבחן לאורך הפרויקט:

1.אנו משווים בין איכויות הלמידה עבור אלגוריתמים נבחרים מהקורס אשר מופיעים כדלקמן.
2.אנו בודקים אילו אלגוריתמים למדו בצורה איכותית ואילו לא ומדוע יש הבדלים בביצועי?
3.אנו בוחנים את שיטת OOD- שבה אנו לוקחים מחלקה חדשה שלא אומנה ובודקים את ההתמודדויות של האלגוריתמים במצב החדש ובוחנים איך הם הגיבו.

תמצית הקוד:

1.בנינו פונקציה אשר מחזירה את המאגר כאשר הוא מחולק ל-train ו-test. בנוסף, ישנה אפשרות להחזיר train ו-test עם מחלקות Out-of-Distribution מובנות כפי שתואר.
2.בנינו פונקציה אשר בודקת את כל השיטות עבור אותו מאגר:

- SVM
- RandomForest
- MLP
- KMeans
- Adaboost
- CNN
- CNN + RandomForest (OOD only)

3.יצרנו מספר פונקציות אשר מוציאות פיצ'רים מן המאגר אשר אותם נשווה:

- **שיטה נאיבית**: שיטוח התמונות ללא הוצאה של פיצ'רים מיוחדים ושימוש בכל פיקסל כפי'צר.
- **מציאת קצוות**: אופרטור prewit - מחשב את הגרדיאנט בכל נקודה ובכך מוצא שינוייסוקצוות.
- **היסטוגרמת גרדיאנטים**: אלגוריתם למציאת אובייקטים - מחשב גרדיאנטים ואתההיסטוגרמה שלהם בצורה קצת שונה מהקצוות שמאפשר תוצאות טובות יותר.
-

כלים:

- ספריית למידה עמוקה tensorflow
- ספריית למידת מכונה sklearn
- ספריית עיבוד תמונות skimage, cv2
- ספריית ייצוג גרפי matplotlib
- ספריית numpy לעבודה מהירה עם מטריצות

מציאת Out-of-Distribution:

בכדי להשוות בין האלגוריתמים השונים בצורה עקבית, חיפשנו שיטה מקובלת למציאת Out of Distribution אשר משותפת לכל האלגוריתמים. השיטה שבחרנו היא יצירת Rejection class - או מחלקת K+1 כאשר K הוא מספר המחלקות שאנו לומדים לעשות עליהם קלסיפיקציה. בזמן האימון, אנו מאמנים את המחלקה הזו לזהות 10 מחלקות בו זמנית מתחומים שונים מהמאגר.

אחד היישומים הוא שימוש ב- RandomForest על הפלט של ה-CNN שלנו - תחת ההנחה- RandomForest יוכל לשפר את הביצועים של רשת הנוירונים במציאת Out-of-Distribution.

תוצאות:

תוצאות מודלים

	RandomForest	Adaboost	SVM	MLP	KMeans	CNN	CNN+RF
Flatten	**22%**	6%	6%	16%	1%	x	x
HoG	19.32%	**7.13%**	**12.33%**	**25%**	0.7%	x	x
Edge	12%	5%	2%	10%	1%	x	x
CNN	x	x	x	x	x	26%	22.67%

תוצאות מודלים + OOD

	RF		ADABOOST		SVM		MLP		KMeans		CNN		CNN + RF	
	ACC	OOD	ACC	OOD	ACC	OOD	ACC	OOD	ACC	OOD	ACC	OOD	ACC	OOD
Naive	**22.29%**	75.10%	9%	29.50%	7.50%	15.60%	18.92%	**43.20%**	1%	0.70%	x		x	
HoG	18%	80.80%	**10.80%**	31.70%	**11.76%**	22.80%	**24.84%**	38.60%	1.00%	0.70%	x		x	
Edge	14.10%	**90.06%**	8.90%	**62.30%**	0.40%	**30%**	11%	31.60%	0.009	0	x		x	
CNN	x	x	x	x	x	x	x	x	x	x	24%	34.8%	24.6%	**42%**

ניתוח התוצאות:

Classification:

ניתן לראות שיפור ניכר עבור Adaboost, SVM, MLP כאשר הוצאנו מן המידע היסטוגרמת גרדיאנטים. לעומת זאת, כאשר ניסינו להוציא קצוות הייתה ירידה בדיוק המודלים. ב-RandomForest לא נראה יתרון ואף הייתה ירידה ע"י היסטוגרמה. התוצאה הטובה ביותר הייתה של הלמידה עמוקה, אשר יכולה בעצמה ללמוד פיצ'רים.

דיוק המודלים מוגבל מטעמים פרקטיים - אימון של SVM עבור מספר רב של דוגמאות אינו יעיל, ולקח שעות רבות אפילו לאחר הגבלתו. את רשת הנוירונים הגבלנו לכמה שכבות בסיסיות שלקונבולוציה שכן רצינו מודל בסיסי ולא בהכרח לרדוף אחר התוצאות האופטימליות ביותר.

Out-of-Distribution Detection:

כפי שניתן לראות, למרות ש-10% מהמידע הועבר למחלקה לא ידועה - הצלחנו לשמר כמעט לחלוטין את דיוק המודל. יכולת תפיסת ה-Out-of-Distribution הייתה גבוהה אם נתייחס לקושי המאגר.

באופן מפתיע, דיוק המודל הכללי לא בהכרח תאם ליכולת OOD - דווקא המודל הגרוע מבניהם(קצוות) הצליח למצוא את המספר המקסימלי של המחלקות הלא ידועות.

אלגוריתם **Kmeans** נמצא לא מתאים **בכלל** בשיטה זו - זאת כיוון שהמחלקות הלא ידועות זרות אחת לשנייה, ולכן האשכול לא ניתן להפרדה.