|  |  |
| --- | --- |
| نام و نام خانوادگی | **سارا رستمی – امین شاهچراغی** |
| شماره دانشجویی | **810۱00۳۵۵ - 810۱۹۹۱۹۶** |
| تاریخ ارسال گزارش | **۱۴۰۱.۰۹.۰۳** |

|  |  |  |
| --- | --- | --- |
|  | **به نام خدا**  **دانشگاه تهران**  **دانشکده‌ مهندسی برق و کامپیوتر** |  |
| **درس شبکه‌های عصبی و یادگیری عمیق**  **تمرین دوم** | | |

**فهرست**

[**پاسخ 1**. **تاثیر تغییر رزولوشن در طبقه‌بندی در شبکه CNN** 1](#_Toc120210557)

[دست‌گرمی: **آشنایی اولیه با** CNN**ها** 1](#_Toc120210558)

[الف) 2](#_Toc120210559)

[ب) 3](#_Toc120210560)

[ج) 3](#_Toc120210561)

[د) 6](#_Toc120210562)

[**پاسخ ۲** **– آشنایی با معماری شبکه CNN** 9](#_Toc120210563)

[۱-۲. **عنوان بخش اول** 9](#_Toc120210564)

**شکل‌ها**

شکل 1- سوال دست‌گرمی 1

شکل 2 - پنج تصویر تصادفی از دیتاست با رزولوشن‌های ۳۲\*۳۲ و ۱۶\*۱۶ و ۸\*۸ 2

شکل 3- نتیجه‌ی تست مدل TOTV با تصاویر ۳۲\*۳۲ 4

شکل 4 - نتیجه‌ی تست مدل TOTV با تصاویر ۲۴\*۲۴ 4

شکل 5- نتیجه‌ی تست مدلTOTV با تصاویر ۱۶\*۱۶ 5

شکل 6 - نتیجه‌ی تست مدل TOTV با تصاویر ۸\*۸ 5

شکل 7- نتیجه‌ی تست مدل TVTV با تصاویر ۳۲\*۳۲ 6

شکل 8 - نتیجه‌ی تست مدل TVTV با تصاویر ۲۴\*۲۴ 7

شکل 9- نتیجه‌ی تست مدل TVTV با تصاویر ۱۶\*۱۶ 7

شکل 10- نتیجه‌ی تست مدل TVTV با تصاویر ۸\*۸ 7

**جدول‌ها**

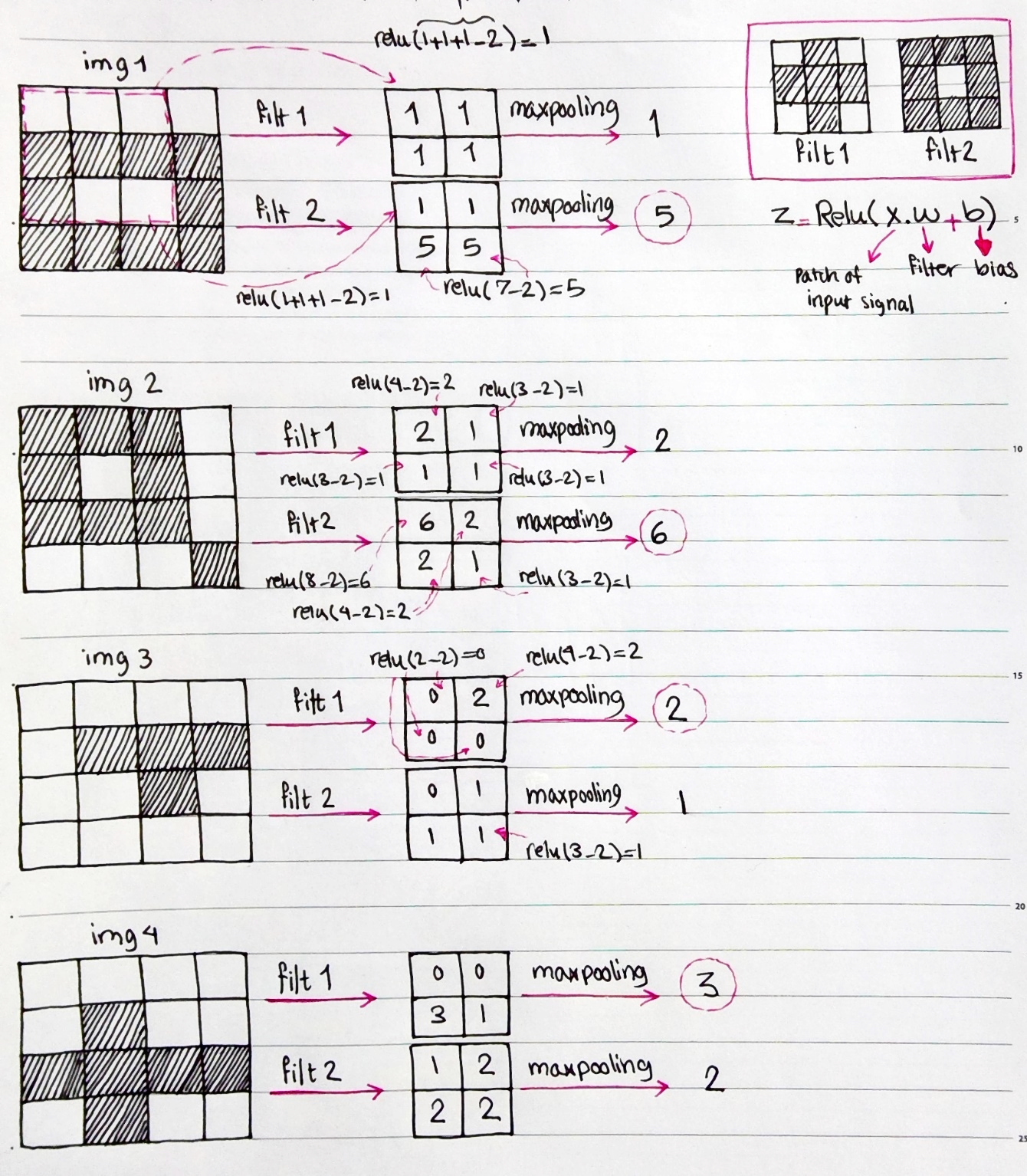
[جدول 1- معماری CNN برای دیتاست CIFAR10 3](#_Toc120201974)

[جدول 2 - مقایسه‌ی نتایج بدست آمده‌ی ما و مقاله در روش TOTV 5](#_Toc120201975)

[جدول 3- مقایسه‌ی نتایج بدست آمده‌ی ما و مقاله در روش TVTV 7](#_Toc120201976)

# **پاسخ 1**. **تاثیر تغییر رزولوشن در طبقه‌بندی در شبکه CNN**

## دست‌گرمی: **آشنایی اولیه با** CNN**ها**

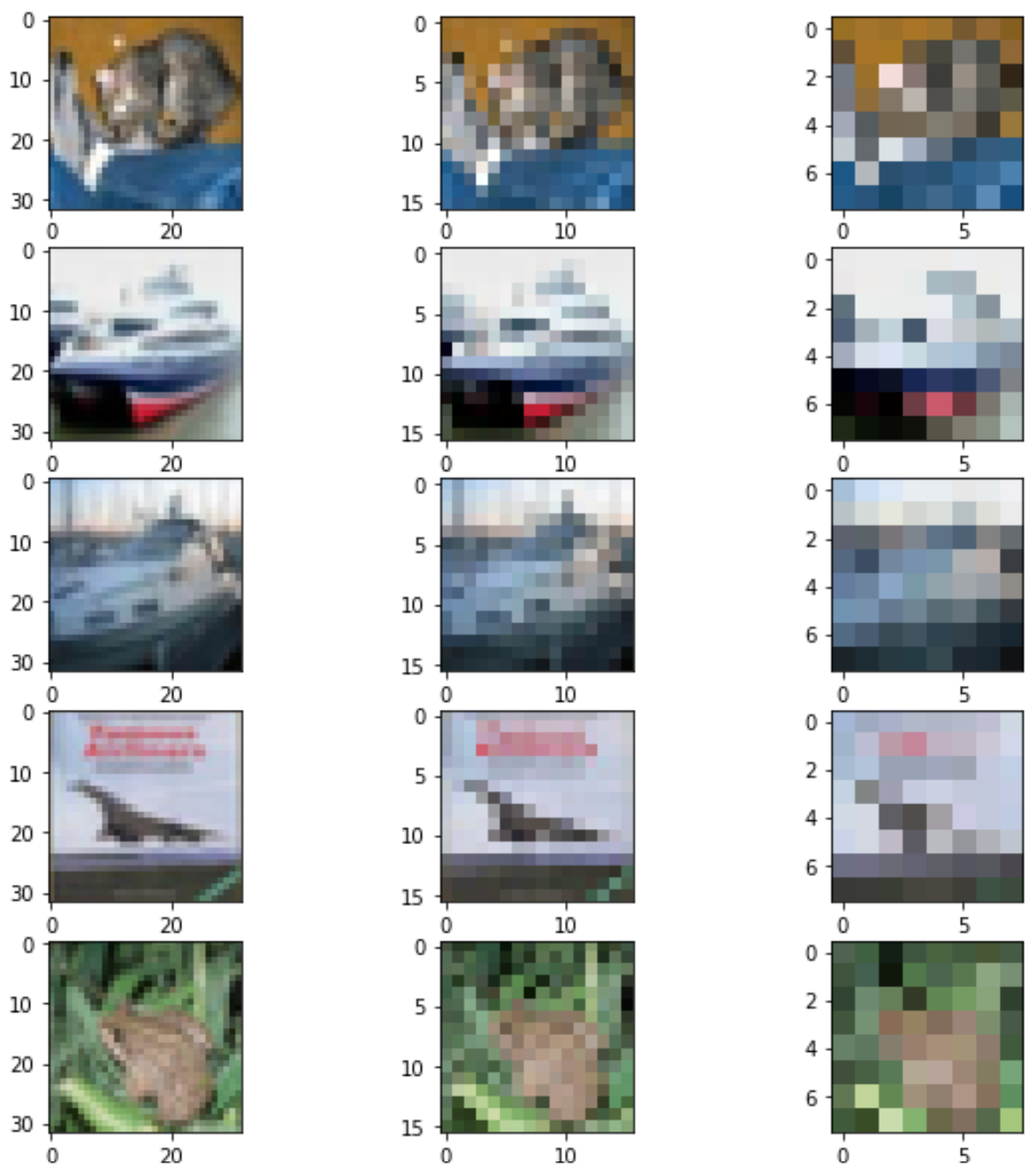


شکل 1- سوال دست‌گرمی

همانطور که در شکل ۱ می‌بینید، نتیجه‌ی maxpooling توسط فیلتر ۲ برای تصاویر شماره ۱ و ۲ بالاتر می‌باشد. به عبارتی فیلتر ۲ بهتر می‌تواند تصاویر کلاس ۱ را طبقه‌بندی کند (زیرا الگوی موجود در فیلتر ۲ در تصاویر کلاس ۱ وجود دارد) و به طور مشابه فیلتر ۱ بهتر قادر به تشخیص تصاویر شماره ۳و ۴ (تصاویر کلاس ۲) می‌باشد چرا که الگوی فیلتر ۱ در این تصاویر وجود دارد.

## الف)

داده‌های test را با کمک تابع resize از کتابخانه‌ی cv2 تغییر اندازه دادیم. سپس ۱۰ تصویر تصادفی (۱۰ تصویر اول داده‌های test) را با رزولوشن‌های ۳۲\*۳۲ و ۱۶\*۱۶ و ۸\*۸ نمایش دادیم. ۵ تصویر اول را در شکل ۲ مشاهده می کنید ( به علت گرفتن فضای بیشتر از اندازه‌ي مانیتور توسط ۱۰ عکس، تنها ۵تا‌ی اول را اسکرین‌شات گرفته و در گزارش آوردیم. نمایش کامل در نوت‌بوک موجود است)



شکل 2 - پنج تصویر تصادفی از دیتاست با رزولوشن‌های ۳۲\*۳۲ و ۱۶\*۱۶ و ۸\*۸

## ب)

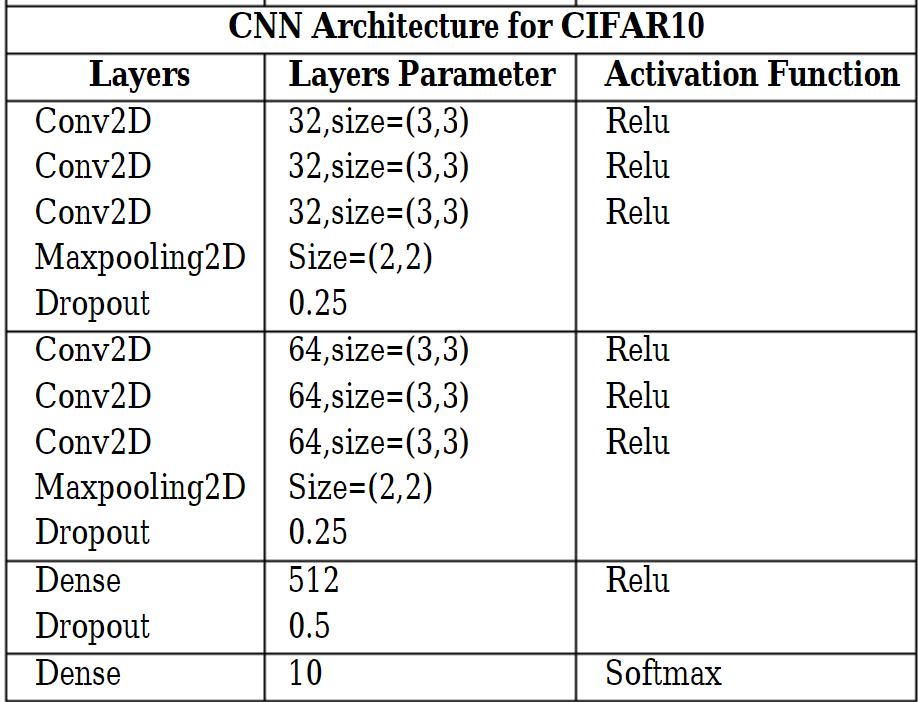
غالب ترین استراتژی برای تقسیم کردن داده ها به ترین ولیدیشن و تست، انجام تقسیم به صورت تصادفی است. البته در همین استراتژی تصادفی چند پارامتر وجود دارد که باید در رابطه با آن تصمیم گیری کرد. به عنوان مثال اگر توزیع کلاس ها تقریباً متوازن است می توان به سادگی از استراتژی رندوم استفاده کرد. اما در صورتی که توزیع کلاس ها یکسان نباشد، اما یادگیری هر چه بهتر پیش بینی همه کلاس ها مد نظر باشد، می توان از استراتژی استریفاید استفاده کرد. علاوه بر همه اینها، باید تصمیم گرفت که چند درصد از داده ها را به هر کدام از ترین، ولیدیشن و اسپلیت اختصاص دهیم. مقادیر رایجی برای این کار وجود دارد که معمولا این مقادیر با توجه به ذات مجموعه داده مثل اندازه، سختی پیش بینی نمونه ها، توزان یا عدم توازن کلاس ها و غیره مشخص می شود. از اعداد رایج برای این کار می توان به درصد 80و 10، 10 و 70، 15 و 15 اشاره کرد. لازم به ذکر است، نسخه هایی مثل گروپ بای هم برای تقسیم وجود دارد، که در شرایط خاصی استفاده می شود.

از آنجایی که دیتاست CIFAR10 یک دیتاست ۱۰ کلاسه متوازن است، از استراتژی رندوم استفاده می‌کنیم.

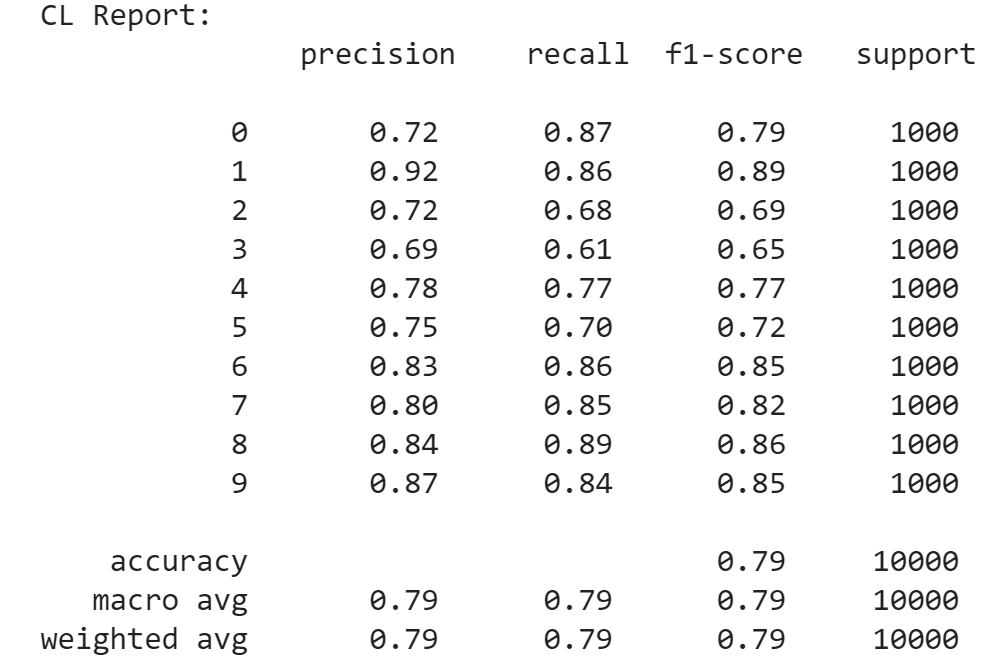
## ج)

برای اینکه بتوانیم تصاویر resize شده را به ورودی شبکه‌ي train شده با تصاویر ۳۲\*۳۲ بدهیم، داده‌های تست resize شده را با درون‌یابی به روش INTER\_CUBE به ۳۲\*۳۲ تغییر سایز دادیم و ذخیره کردیم. حال CNN را با معماری ذکر شده در مقاله مطابق جدول ۱ طراحی کردیم. همچنین از SGB به عنوان optimizer و از cross\_entropy به عنوان تابع loss استفاده کردیم.

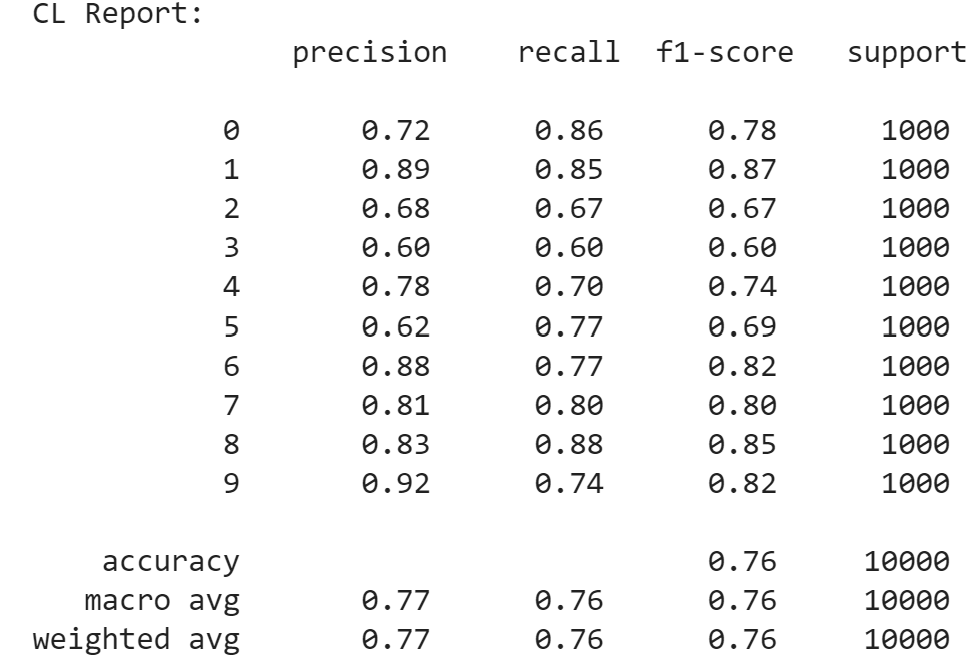
جدول 1- معماری CNN برای دیتاست CIFAR10



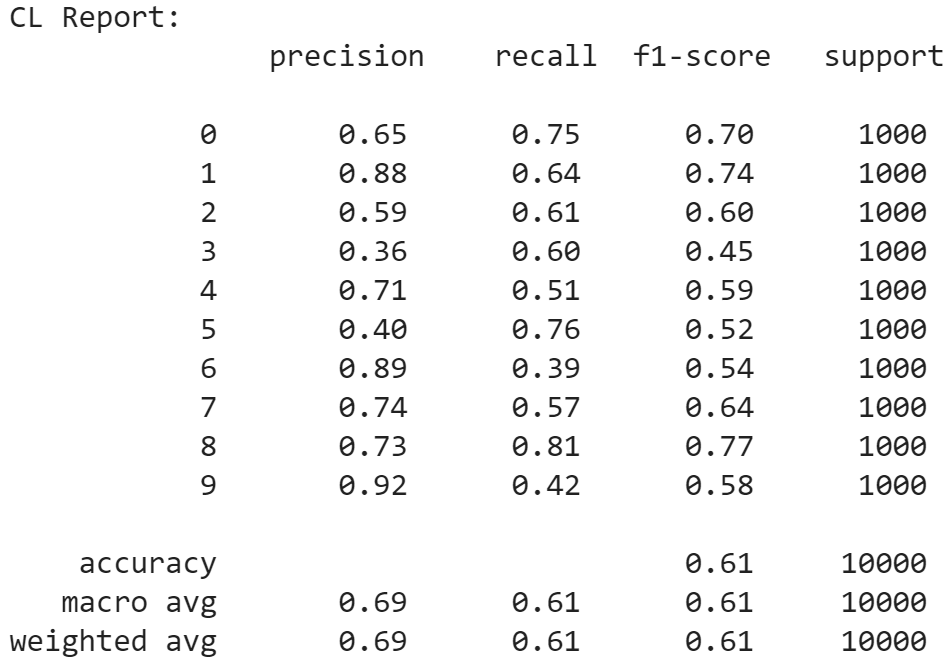
طبق روش ارا‌ئه شده‌ی مقاله برای TOTV ابتدا شبکه را با تصاویر ۳۲\*۳۲ طی ۱۰۰ epoch، آموزش داده و سپس با تصاویر ۳۲\*۳۲ و ۲۴\*۲۴ و ۱۶\*۱۶ و ۸\*۸ تست کردیم. نتیجه‌ی تست با هر سایز از تصاویر را در شکل‌های ۳ و ۴ و ۵ و ۶ مشاهده می‌کنید. در نهایت برای مقایسه‌ی نتایج بدست آمده‌ی ما و مقاله، آن‌ها در جدول ۲ نمایش دادیم.



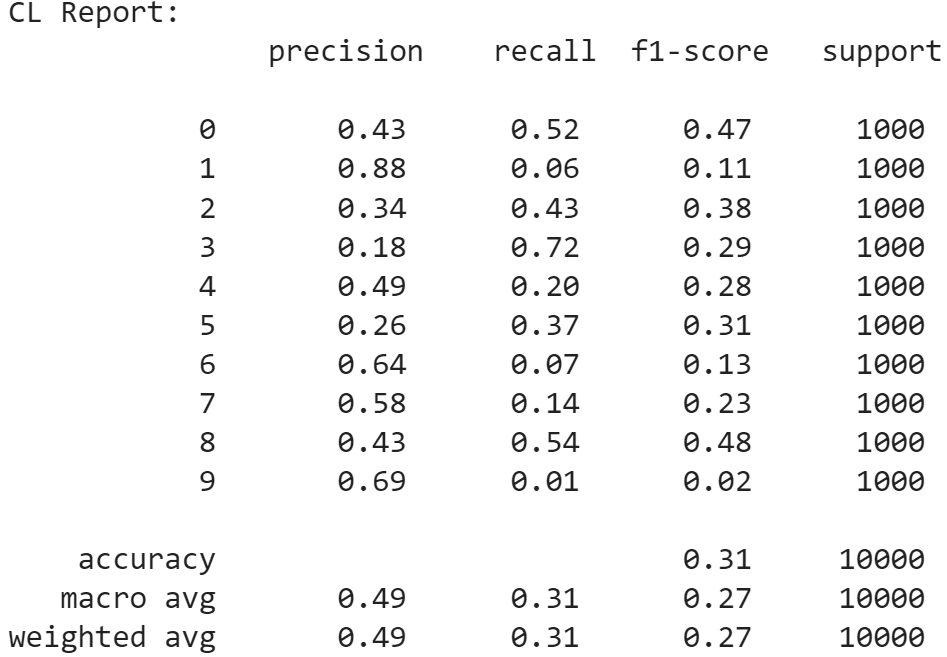
شکل 3- نتیجه‌ی تست مدل TOTV با تصاویر ۳۲\*۳۲



شکل 4 - نتیجه‌ی تست مدل TOTV با تصاویر ۲۴\*۲۴



شکل 5- نتیجه‌ی تست مدلTOTV با تصاویر ۱۶\*۱۶



شکل 6 - نتیجه‌ی تست مدل TOTV با تصاویر ۸\*۸

مقایسه‌ی نتایج بدست آمده توسط ما و نتایج مقاله برای روش TOTV را در جدول ۲ مشاهده می‌کنید.

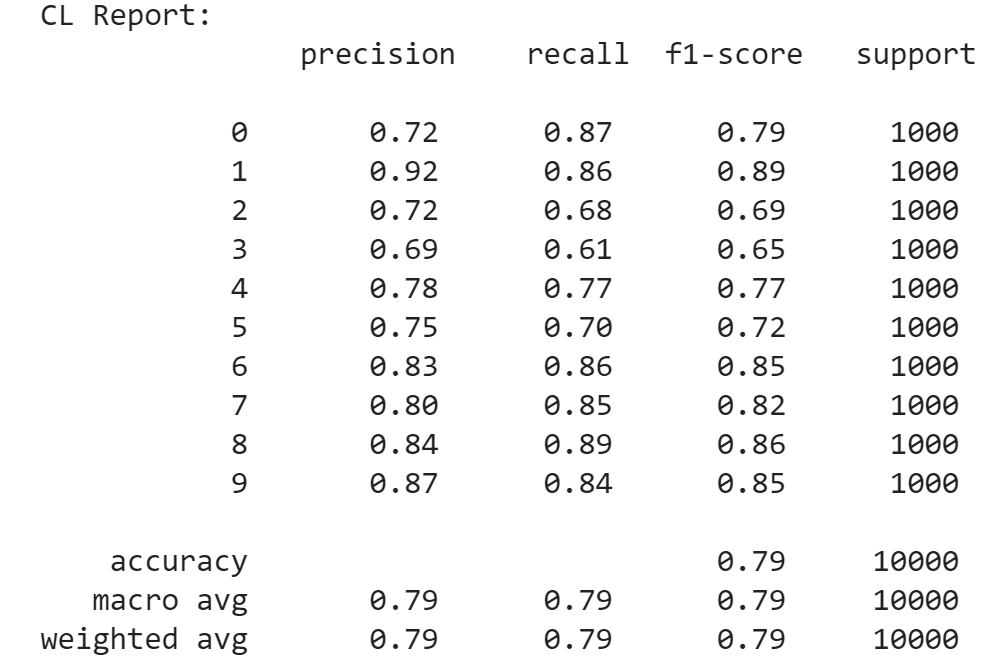
جدول 2 - مقایسه‌ی نتایج بدست آمده‌ی ما و مقاله در روش TOTV

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **CIFAR10**  **Dataset**  **Resolution** | **Accuracy** | | **Precision** | | **F1 Score** | |
| paper | ours | paper | ours | paper | ours |
| **32\*32** | 0.8752 | 0.79 | 0.87652 | 0.79 | 0.87548 | 0.79 |
| **24\*24** | 0.6409 | 0.76 | 0.72365 | 0.77 | 0.65320 | 0.76 |
| **16\*16** | 0.3166 | 0.61 | 0.48415 | 0.69 | 0.29897 | 0.61 |
| **8\*8** | 0.1855 | 0.31 | 0.27090 | 0.49 | 0.13986 | 0.27 |

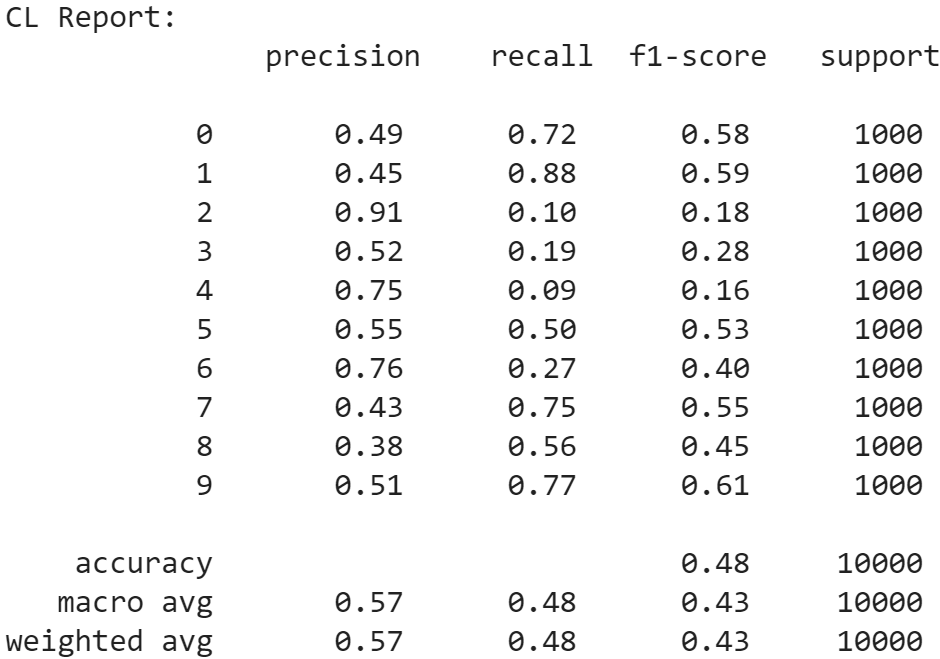
همانطور که در جدول ۲ مشاهده می‌کنید، نتایج بدست آمده‌ی ما به غیر از تصاویر ۳۲\*۳۲ برای سایر داده‌های تست، نسبت به نتایج مقاله بهتر می‌باشد. همچنین طبق انتظار، تست با داده‌های ۳۲\*۳۲ نتایج بهتری نسبت به سایر رزولوشن‌ها به ما می‌دهد. چرا که داده‌های با رزولوشن کمتر دچار مقداری information loss شده‌اند.

## د)

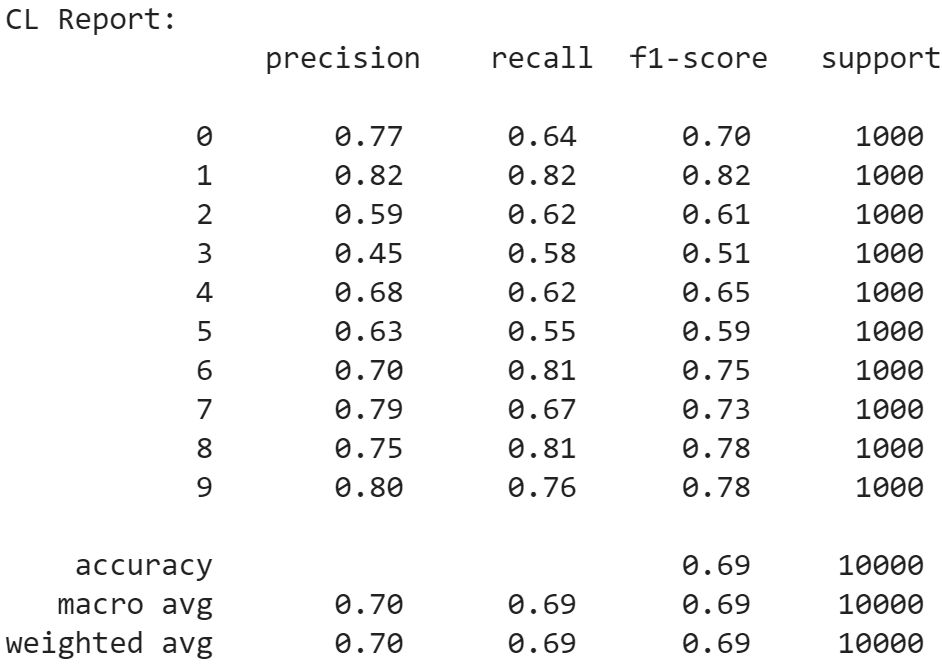
معماری شبکه همانند بخش ج مطابق جدول ۱ می‌باشد. طبق روش ارا‌ئه شده‌ی مقاله برای TVTV ۴ شبکه را به ترتیب با تصاویر ۳۲\*۳۲ و ۲۴\*۲۴ و ۱۶\*۱۶ و ۸\*۸ طی ۱۰۰ epoch، آموزش داده و سپس آن‌ها را به‌ترتیب با تصاویر ۳۲\*۳۲ و ۲۴\*۲۴ و ۱۶\*۱۶ و ۸\*۸ تست کردیم. (دقت شود که شبکه‌ی آموزش یافته با تصاویر ۳۲\*۳۲ و تست شده با تصاویر ۳۲\*۳۲ در کد قسمت ج موجود است). نتیجه‌ی تست هر با هر سایز از تصاویر را در شکل‌های۷و ۸ و ۹ و ۱۰ مشاهده می‌کنید. در نهایت برای مقایسه‌ی نتایج بدست آمده‌ی ما و مقاله، آن‌ها در جدول ۳ نمایش دادیم.



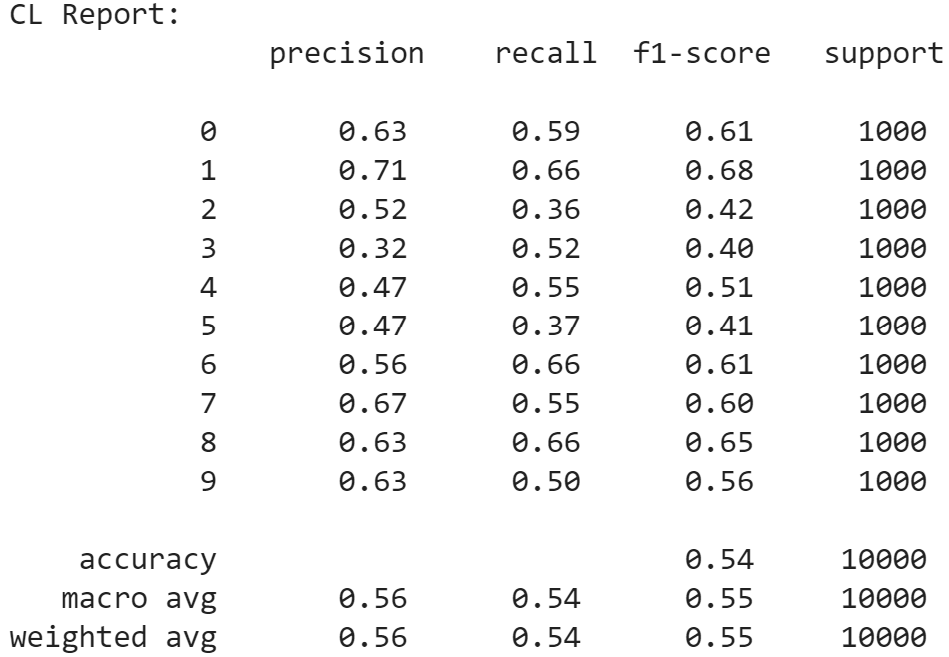
شکل 7- نتیجه‌ی تست مدل TVTV با تصاویر ۳۲\*۳۲



شکل 8 - نتیجه‌ی تست مدل TVTV با تصاویر ۲۴\*۲۴



شکل 9- نتیجه‌ی تست مدل TVTV با تصاویر ۱۶\*۱۶



شکل 10- نتیجه‌ی تست مدل TVTV با تصاویر ۸\*۸

مقایسه‌ی نتایج بدست آمده توسط ما و نتایج مقاله برای روش TOTV را در جدول ۲ مشاهده می‌کنید.

جدول 3- مقایسه‌ی نتایج بدست آمده‌ی ما و مقاله در روش TVTV

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **CIFAR10**  **Dataset**  **Resolution** | **Accuracy** | | **Precision** | | **F1 Score** | |
| paper | ours | paper | ours | paper | ours |
| **32\*32** | 0.8752 | 0.79 | 0.87652 | 0.79 | 0.87548 | 0.79 |
| **24\*24** | 0.6204 | 0.48 | 0.70501 | 0.57 | 0.63220 | 0.43 |
| **16\*16** | 0.4233 | 0.69 | 0.62030 | 0.70 | 0.40654 | 0.69 |
| **8\*8** | 0.3020 | 0.54 | 0.54599 | 0.56 | 0.24262 | 0.55 |

با مشاهده‌ی جدول ۳، می‌بینیم که با مدل‌های ۱۶\*۱۶ و ۸\*۸ نتایج بهتری نسبت به مقاله گرفتیم. دلیل آن کمتر بودن تعداد پیکسل‌ها ( ویژگی‌ها) بوده که منجر به زودتر همگرا شدن مدل در ۱۰۰ ایپاک می‌شود. با توجه به دقت و loss مدل هنگام یادگیری با داده‌ی train، به نظر می‌رسد که با افزایش تعداد ایپاک‌ها، می‌توان به مدل بهتری رسید. با این حال بهترین accuracy، Precision و f1 Score را با مدل train و test شده توسط تصاویر ۳۲\*۳۲داریم که دلیل آن حفظ ویژگی‌های بیشتری از هر تصویر و در نتیجه توانایی بالاتر در یادگیری آنها می‌باشد که منجر به پیش‌بینی‌ بهتر و دقیق‌تر داده‌های تست می‌شود.

با مقایسه‌ی جدول ۲ و۳ می‌بینیم که به طور کلی نتایج بدست آمده ‌ی ما با مدل TOTV بهتر از مدل TVTV می‌باشد. پس می‌توان نتیجه گرفت که آموزش مدل با تصاویر با رزولوشن بالاتر می‌تواند منجر به مدل‌های قدرتمندتری شود که می‌توانند حتی تصاویر با رزولوشن پایین‌تر را بهتر دسته‌بندی کنند.

# **پاسخ ۲** **– آشنایی با معماری شبکه CNN**

## ۱-۲. **عنوان بخش اول**

متن نمونه

.