به نام خدا

دانشگاه تهران

پردیس دانشکده‌های فنی

دانشکده برق و کامپیوتر

یادگیری عمیق با کاربرد در بینایی ماشین و پردازش صوت

پروژه‌ی امتیازی

گروه ۴

نام و نام خانوادگی:

سارا جاهد آزاد ۸۱۰۶۹۹۱۴۹

محمد مهدی مهمانچی ۸۱۰۱۹۹۲۸۷

عرفان میرزایی ۸۱۰۱۹۹۲۸۹

مرداد ماه ۱۴۰۰

فهرست

فهرست اشکال

فهرست جداول

چکیده

در این پروژه، مقاله‌ی [1] پیاده‌سازی‌شد. شبکه‌ی معرفی‌شده در این مقاله می‌تواند تصاویر جدیدی را با ویژگی‌های[۱] دلخواه تولیدکند. این مقاله برای این امر، از یک ساختار دو مرحله‌ای بهره‌برده‌است. در قسمت اول ابتدا شبکه‌ی تولیدکننده[۲] تصاویری را تولید می‌نماید و سپس در ادامه یک شبکه‌ی طبقه‌بند[۳] به طبقه‌بندی ویژگی های موجود در تصویر ایجاد شده می‌پردازد. ساختار مورد استفاده در این مقاله شبیه به شبکه های مولد تخاصمی[۴] است با این تفاوت که در این مقاله، تولیدکننده و طبقه‌بند به همراه یکدیگر کار می‌کنند و نه برخلاف یکدیگر.

این شبکه برای مجموعه دادگان[۵] تصاویر چهره (KDEF) پیاده‌سازی‌شد. در مرحله‌ی اول تصاویری با احساسات متفاوت و هم چنین با چرخش‌های متفاوت در بخش شبکه‌ی تولیدکننده تولیدشدند، و در ادامه در قسمت شبکه‌ی طبقه‌بند، هویت فرد، احساس او و جهت چرخش تصویر او مورد طبقه‌بندی قرار گرفت. این شبکه برای مجموعه دادگان اعداد دست‌نویس(QMNIST)، نیز پیاده‌سازی‌شد. در این حالت ویژگی‌های مورد طبقه‌بندی، فرد نویسنده‌ی آن عدد، خود عدد نوشته‌شده، رنگ عدد و جهت چرخش آن بودند.

در انتها از شبکه‌ی پیاده‌سازی‌شده برای مجموعه‌ی دادگان KDEF، برای تولید تصاویر جدید به منظور افزودن دادگان[۶] استفاده‌شد، و این داده‌ها برای آموزش شبکه‌ای با دقت بیشتر به‌کارگرفته‌شدند.

---

۱. Features
۲. Generator
۳. Classifier
۴. Generative Adversarial Network(GAN)
۵. Dataset
۶ Data Augmentation

سوال ۱

۱. مقدمه و هدف مقاله

در این پروژه به پیاده‌سازی مقاله‌ی Generative Cooperative Net for Image Generation and Data Augmentation پرداخته شده است. ساختن یک مدل خوب برای تولید تصاویر، یکی از مسئله‌های مطرح در بینایی ماشین است. به طور کلی در این مقاله، یک روش تولیدی جدید مطرح شده که می‌تواند مفاهیم سطح بالا را یاد بگیرد و تصاویر جدید با کیفیت بالا که ویژگی‌های[۱] دلخواه ما را دارند را تولید کند. آزمایش‌های مقاله روی مجموعه دادگان[۲] اعداد دستنویس (QMNIST) و احساسات چهره (KDEF) انجام شده است که برای این کار، شبکه‌ای متشکل از دو بخش تولیدکننده[۳] و طبقه بند[۴] استفاده شده است. همچنین مقاله از شبکه‌ی ایجاد شده به منظور افزودن داده[۵] نیز بهره برده است. در این پروژه ما هر سه مرحله ذکر شده را پیاده‌سازی کردیم و به مقایسه و بررسی نتایج بدست آمده با نتایج ذکر شده در مقاله پرداخته‌ایم. در ادامه به توضیح جزئیات پیاده سازی هر یک از مراحل می‌پردازیم؛ بدین منظور ابتدا به شرح تولید تصاویر احساسات چهره(KDEF) می‌پردازیم ، سپس به آزمایش روی مجموعه دادگان اعداد دستنویس (QMNIST) پرداخته‌می‌شود و در انتها به طبقه بندی احساسات موجود در تصاویر صورت به کمک عکس‌های افزوده شده[۶] توسط شبکه مرحله قبل می‌پردازیم.

۲. پیاده سازی تولید تصاویر احساسات چهره

مقاله‌ی مربوط به این پروژه، به پیاده‌سازی کاربرد شبکه‌ی خود برای تولید تصاویر از چهره افراد در حالت های گوناگون شامل احساسات متفاوت و هم چنین چرخش‌های متفاوت تصویر پرداخته‌است. بدین منظور این مقاله، از مجموعه‌ی داده‌ی KDEF بهره‌برده‌است که در ادامه به آماده‌سازی داده‌ها برای به کارگیری در شبکه پرداخته می‌شود.

۲,۱. آماده‌کردن مجموعه داده:

در مقاله‌ی این پروژه از مجموعه داده KDEF استفاده شده است. این مجموعه شامل تصاویر چهره ۷۰ فرد (۳۵ زن و ۳۵ مرد) است که از پنج زاویه مختلف تصویر برداری شده است. در مقاله فقط از زاویه مستقیم (تمام

---

۱. Features.
۲. Dataset.
۳. Generator.
۴. Classifier.
۵. Data Augmentation.
۶. Augmented Images.

رخ) استفاده شده است. همچنین هر فرد ۷ حالت گوناگون چهره به خود گرفته است که در مقاله از چهار حالت عادی[۱]، خوشحال[۲]، منزجر[۳] و متعجب[۴] استفاده شده است. در ضمن از هر فرد در دو نوبت عکس گرفته شده است که ما فقط از عکس های نوبت اول استفاده کرده ایم. نام هر تصویر یک رشته به طول ۷ یا ۸ می باشد که شامل کلیه‌ی اطلاعات لازم برای آن تصویر می باشد. برای نمونه نام یک تصویر به صورت AF06SUS.JPG می باشد که به کمک آن اطلاعات زیر درباره تصویر به دست می آید:

- حرف اول نشان دهنده نوبت تصویر برداری است: A یا B
- حرف دوم نشان دهنده جنسیت است: F یا M
- حرف های سوم و چهارم نشان دهنده شماره فرد است: از 00 تا ۳۵
- حرف های پنجم و ششم مشخص کننده حالت چهره است: AF, AN, DI, HA, NE, SA, SU
- حرف های هفتم و هشتم(در صورت وجود) زاویه تصویر برداری را نشان می دهد: FL, HA,S,HR,FR

در این بخش مطابق روشی که مقاله ذکر کرده است با استفاده از پکیج opencv و الگوریتم CascadeClassifier آن در هر تصویر چهره فرد را شناسایی می کنیم و عکس را برش[۵] می زنیم تا فقط شامل صورت باشد. در نهایت ابعاد این تصویر را به اندازه ۱۵۸ در ۱۵۸ در می آوریم. در شکل زیر یک نمونه تصاویر پیش از و پس از اعمال پیش پردازش آورده شده است.

شکل ۱- تصویر پس از انجام برش و تشخیص چهره

شکل ۲- تصویر موجود در مجموعه دادگان بدون پیش پردازش

۱.Neutral
۲ .Happy
۳ .Disgusted
۴.Surprise
۵.Crop

در ادامه هر یک از تصاویر بدست آمده را در سه زاویه ۹۰ و ۱۸۰ و ۲۷۰ می چرخانیم[۱]. در نهایت تصویر اصلی و سه چرخش آن را نسبت به محور تقارن چهره بازتاب[۲] می کنیم؛ بدین ترتیب از هر تصویر اصلی ۷ تصویر جدید به‌دست‌آورده‌ایم و مجموعه داده را غنی تر کرده‌ایم.

در مقاله دو دسته نتایج به عنوان خروجی آورده شده است که این دو دسته نیازمند دو نوع دادگان آزمون می باشند. به همین علت دادگان آزمون به دو صورت زیر ساخته شده‌اند:

چهار نفر را به صورت تصادفی انتخاب می کنیم، برای هر کدام یک حالت چهره و یک زاویه چرخش را باز هم به صورت تصادفی انتخاب می کنیم، تصویر مربوط به این فرد و حالت چهره و زاویه چرخش (همراه با بازتاب) را در مجموعه تست قرار می دهیم.(مجموعا ۸ تصویر)

دو نفر را به صورت تصادفی انتخاب می کنیم، برای هر کدام یک حالت چهره را باز هم به صورت تصادفی انتخاب می کنیم، تصویر مربوط به این فرد و حالت چهره را به ازای تمام زوایای چرخش(همراه با بازتاب) در مجموعه تست قرار می دهیم.(مجموعا ۱۶ تصویر)

بدین صورت مجموعه دادگان آزمون شامل ۲۴ تصویر ساخته می شود.

۲,۲. معماری شبکه

شبکه‌ی این مقاله از دو بخش عمده تشکیل شده است:

۱- تولیدکننده‌ی عکس

۲- طبقه‌بند

که در ادامه هر یک از این بخش ها به تفصیل توضیح داده می‌شوند.

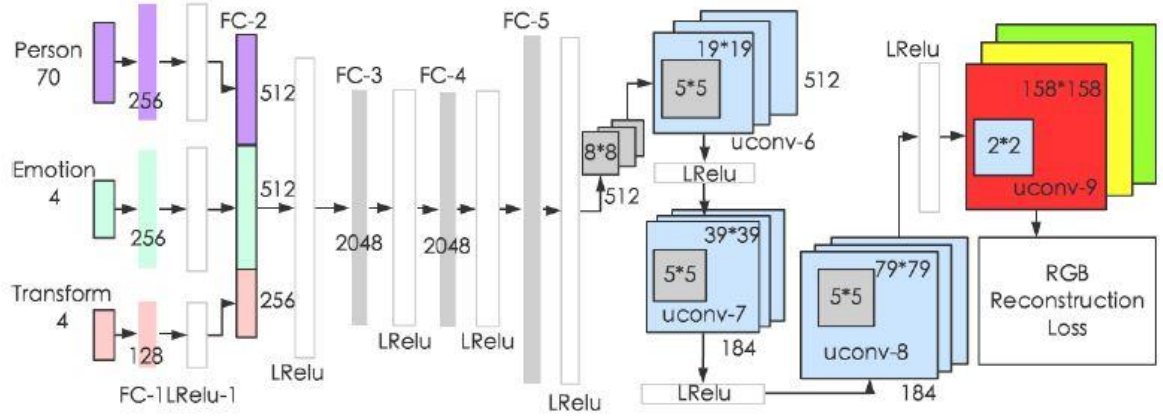
---

۱. Rotate

۲.Mirror

۲,۲,۱. تولیدکننده‌ی عکس

قسمت نخست شبکه برای تولید تصاویر به کار می‌رود. این بخش سه نوع ورودی دارد؛ بردار مربوط به هویت شخص، بردار مربوط به احساس شخص و بردار مربوط به چرخش تصویر فرد. این سه بردار به صورت وان‌هات[۱] از سه ورودی وارد شبکه می‌گردند، از لایه‌های تمام-متصل[۲] مربوط به خود عبور می‌کنند و با هم پیوست[۳] می‌شوند، سپس پس از عبور از چند لایه‌ی تمام-متصل دیگر و پس از آن چند لایه‌ی پیچشی[۴] دیگر، در نهایت یک خروجی به ابعاد 158*158 با سه کانال به دست خواهد آمد که با تصویر اصلی مقایسه می‌گردد(یکی از بخش‌های تابع هزینه در این قسمت محاسبه می‌شود؛ یعنی مجموع خطای مربعات خروجی تولیدکننده با عکس‌های اصلی آموزش). معماری بخش تولیدکننده طبق معماری پیشنهادی در مقاله‌ی [4] بوده است. معماری دقیق این بخش در ادامه مشاهده می‌گردد.



شکل ۳- معماری شبکه تولیدکننده

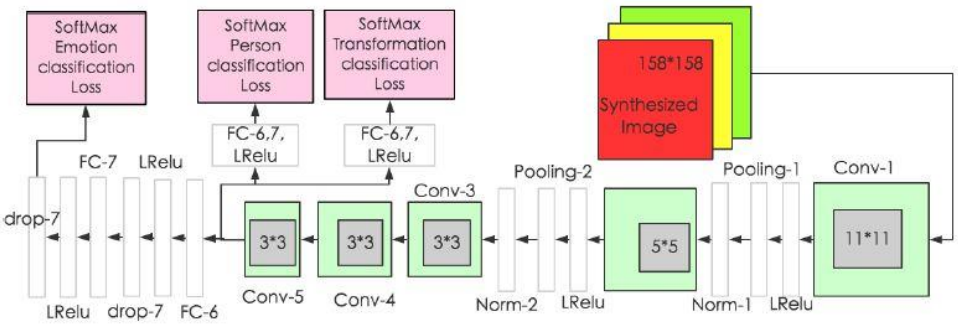
[۱] One Hot
[۲] . Fully-Connected
[۳] .Concatenate
[۴] Convolutional Layers

جدول ۱-ساختار شبکه‌ی تولیدکننده

Layer name	Input Feature Size	Kernel Size	Padding	Stride	Output Feature Size
Person FC 1	70	-	-	-	256
Leaky ReLU (negative slope=0.1)					
Person FC 2	256	-	-	-	512
Emotion FC 1	4	-	-	-	256
Leaky ReLU (negative slope=0.1)					
Emotion FC 2	256	-	-	-	512
Transform FC 1	4	-	-	-	128
Leaky ReLU (negative slope=0.1)					
Transform FC 2	128	-	-	-	256
Then		these three outputs		being concatenated	
Leaky ReLU (negative slope=0.1)					
FC 3	1280	-	-	-	2048
Leaky ReLU (negative slope=0.1)					
FC 4	2048	-	-	-	2048
Leaky ReLU (negative slope=0.1)					
FC 5	2048	-	-	-	16384
Leaky ReLU (negative slope=0.1)					
Reshape to 256*8*8					
deConv 6	256	5	0	2	512
Leaky ReLU (negative slope=0.1)					
deConv 7	512	5	1	2	184
Leaky ReLU (negative slope=0.1)					
deConv 8	184	5	1	2	184
Leaky ReLU (negative slope=0.1)					
deConv 9	184	2	0	2	3

۲،۲،۲. طبقه‌بند

خروجی تولیدکننده‌ی عکس به عنوان ورودی وارد طبقه‌بند می‌شود. قسمت پیچشی[۱] این شبکه از معماری Alexnet[5] الهام گرفته‌شده است؛ با این تفاوت که گام[۲] در لایه‌ی اول به ۲ کاهش یافته (زیرا دادگان استفاده شده در این مقاله ابعاد کوچک‌تری نسبت به مجموعه دادگانی دارد که Alexnet[5] در اصل با آن آموزش پیدا کرده است). همچنین لایه‌های local response normalization به قسمت پیچشی این مقاله اضافه شده‌اند که در مقاله‌ی اصلی Alexnet نیستند. طی بررسی‌های انجام‌شده، جایگزینی لایه‌های batch normalization با LRN پاسخ بهتری به دست می‌دهد. سپس سه قسمت متشکل از لایه‌های تمام-متصل، هر یک برای طبقه‌بندی اشخاص، احساسات و چرخش به انتهای بخش پیچشی شبکه اضافه می‌گردند. در ادامه، معماری بخش طبقه‌بند با جزئیات بیشتر آورده شده است.



شکل**4** - معماری شبکه طبقه بند

جدول ۲-ساختار شیکه‌ی طبقه‌بند

Layer	Input Feature Size	Kernel Size	Padding	Stride	Output Feature Size
Conv 1	3	11	2	2	64
Leaky ReLU (negative slope=0.2)					
Batch Normalization					
Maxpool layer 1	64	3	2	0	64
Conv 2	64	5	2	1	192
Leaky ReLU (negative slope=0.2)					
Batch Normalization					

۱. Convolutional

۲. Stride

Maxpool layer 2	192	3	2	0	192
Conv 3	192	3	1	1	384
Leaky ReLU (negative slope=0.2)					
Conv 4	384	3	1	1	256
Leaky ReLU (negative slope=0.2)					
Conv 5	256	3	1	1	256
Leaky ReLU (negative slope=0.2)					
Maxpool layer 5	256	3	2	0	256
Flattening Layer					
FC 6	16384	–	–	–	4096
Leaky ReLU (negative slope=0.2)					
Dropout(0.5)					
FC 7	4096	–	–	–	4096
Leaky ReLU (negative slope=0.2)					
Dropout(0.5)					
FC 8	4096	–	–	–	70
FC 6	16384	–	–	–	4096
Leaky ReLU (negative slope=0.2)					
Dropout(0.5)					
FC 7	4096	–	–	–	4096
Leaky ReLU (negative slope=0.2)					
Dropout(0.5)					
FC 8	4096	–	–	–	4
FC 6	16384	–	–	–	4096
Leaky ReLU (negative slope=0.2)					
Dropout(0.5)					
FC 7	4096	–	–	–	4096
Leaky ReLU (negative slope=0.2)					
Dropout(0.5)					
FC 8	4096	–	–	–	4

۲,۳. تابع هزینه[۱]

تابع هزینه برای این شبکه به صورت زیر تعریف می‌گردد:

$$Loss: Euclidian\ pixel\ to\ pixel\ loss + 10 * Crossentropy\ Loss_{person} + 10 * Crossentropy\ Loss_{emotion} + 10 * Crossentropy\ Loss_{transformation}$$

تابع هزینه اقلیدسی، در واقع تابع هزینه‌ای است که بین تصاویر تولید شده در بخش تولیدکننده و تصاویر واقعی محاسبه می‌شود.

نمایش کلی تابع هزینه به صورت زیر خواهد بود:

$$\frac{1}{N}\{\sum_{i=1}^{N}\left\|Pixel_r^i - Pixel_s(f_{i_1},\dots,f_{iM})^i\right\|^2 - [\sum_{i=1}^{N}\sum_{f=1}^{M}\sum_{j=1}^{K_f} Weight_f\{y_f^i = j\}\log\frac{e^{\theta_{f_j}^T x_f^i}}{\sum_{l=1}^{K_f} e^{\theta_{f_l}^T x_f^i}}]\}$$

در عبارت بالا N تعداد تصاویر، M تعداد ویژگی‌های هر عکس و K_f بُعد کلاس برای یک ویژگی خاص f است. در فرمول بالا weight برابر با 10 قرار داده می‌شود.

۲,۴. آموزش شبکه

برای آموزش شبکه، از مجموعه ابرپارامترهای[۲] زیر استفاده کرده‌ایم:

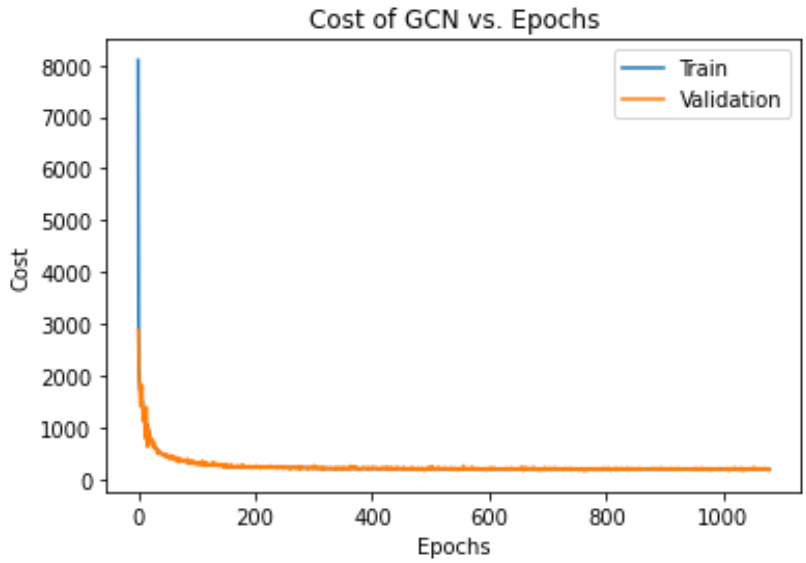
جدول ۳- ابرپارامترهای مورد استفاده در آموزش شبکه

Optimizer	Adam
Learning rate	0.0002
Learning rate decay:	after every 150 epochs it is divided by 2
β_1	0.9
β_2	0.99
ε	10^{-8}
batch size	64
epochs	1100

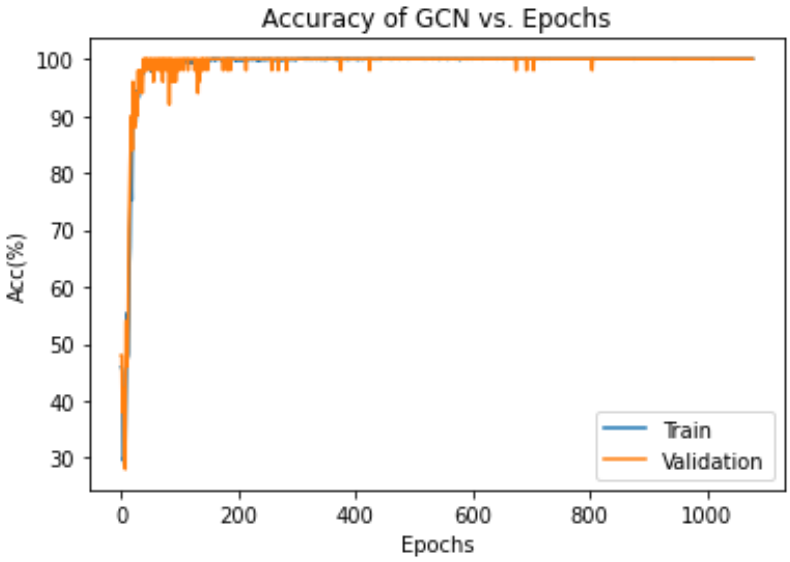
۱. Loss Function.

۲. Hyper-parameters.

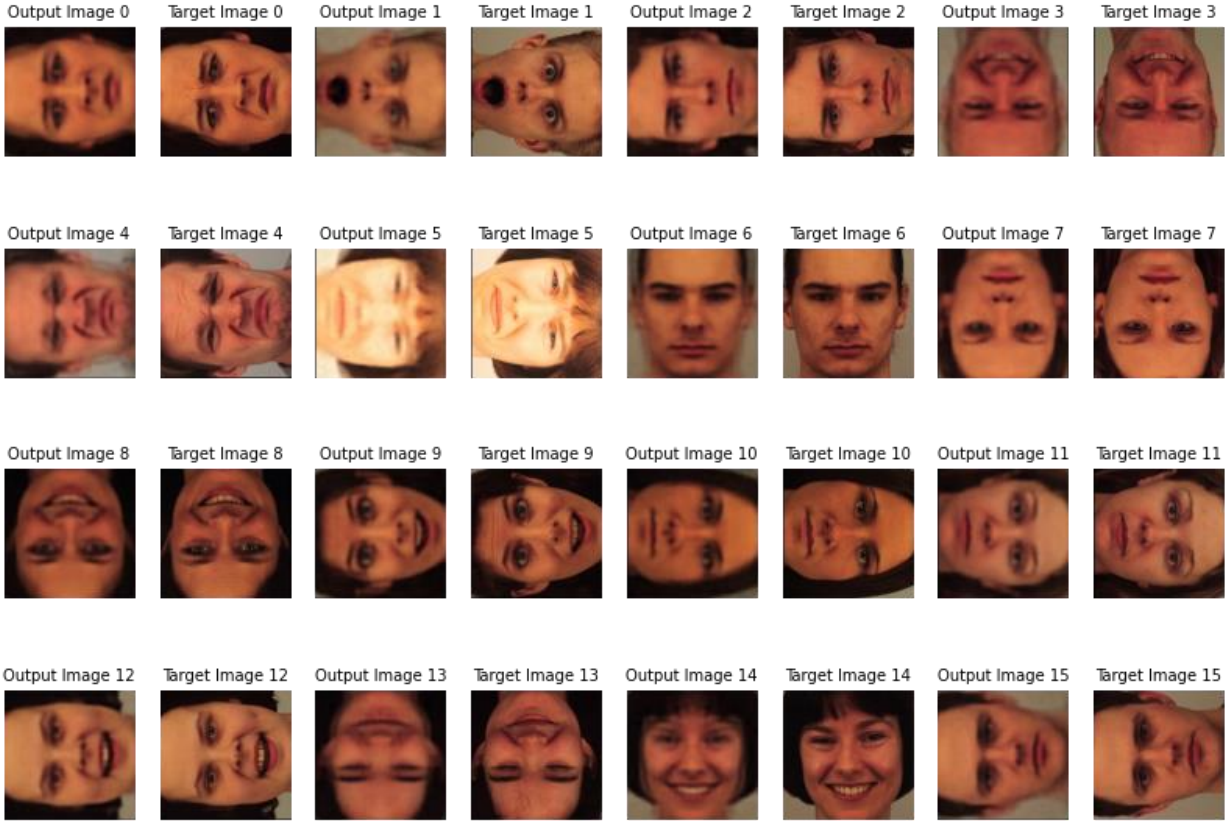
۲,۵. نتایج



شکل ۵- نمودار تابع هزینه در حین آموزش بر دادگان **KDEF**



شکل ۶- نمودار دقت طبقه بند احساسات در حین آموزش بر دادگان **KDEF**



شکل ۷- تصاویر تولید شده به همراه تصاویر اصلی از مجموعه دادگان آموزش **KDEF**

شکل ۹- تصاویر تولید شده(سطح-پایین) به همراه تصاویر اصلی از مجموعه دادگان آزمون **KDEF**

شکل ۸- تصاویر تولید شده(سطح-بالا) به همراه تصاویر اصلی از مجموعه دادگان آزمون **KDEF**

۲,۶. نتیجه‌گیری:

همان طور که نتایج بالا نشان می‌دهد شبکه توانسته است ویژگی های سطح-پایین[۱] مانند چرخش تصاویر دیده شده و هم چنین ویژگی های سطح-بالا[۲] مانند احساسات موجود در چهره‌ی افراد را به خوبی یاد بگیرد. از مقایسه‌ی تصاویر ایجاد شده توسط شبکه بر روی دادگان آموزش و ارزیابی می‌توان تا حدی تفاوت عملکرد را حس کرد که این امر به خاطر کوچک بودن مجموعه دادگان این شبکه با توجه به بزرگ بودن شبکه تا حدی توجیه پذیر است. هم چنین همان طور که مشاهده می‌شود دقت طبقه بندی احساسات، بر روی مجموعه دادگان آموزش و تایید، تقریبا از همان ابتدای کار به ۱۰۰ درصد رسیده است.

۳. پیاده سازی تولید تصاویر اعداد دست‌نویس

مقاله‌ی مربوط به این پروژه، به پیاده‌سازی کاربرد شبکه‌ی خود برای اعداد دست‌نویس نیز پرداخته است. طبق ادعای این مقاله، مجموعه‌ی دادگان مورد استفاده در این پژوهش ، mnist بوده است. باری با بررسی کدهای ارائه‌شده توسط نویسندگان[2] ، این مسئله دریافت شد که این قسمت از پروژه با کمک دادگان qmnist پیاده‌سازی شده است، چرا که فاکتور دیگری تحت عنوان فرم[۳] یا استایل[۴] در برچسب‌های شبکه دخیل‌شده که دادگان mnist فاقد آن هستند. با بررسی بیشتر مشخص شد که دادگان به‌کار رفته در این تحقیق qmnist هستند. در ادامه به آماده‌سازی داده‌ها برای به کارگیری در شبکه پرداخته می‌شود.

۳,۱. آماده‌کردن مجموعه دادگان:

برای بارگذاری داده‌ها از روش به کار رفته در منبع [3] استفاده شده است.

دادگان این مجموعه شامل عکس‌هایی سیاه و سفید از اعداد دست‌نویس است که توسط افراد مختلفی نوشته شده و هر کدام از این نویسندگان با شماره‌ای در برچسب[۵] این تصاویر قابل شناسایی هستند. حالت کلی برچسب این تصاویر، از هشت بُعد تشکیل شده و به صورت زیر است:

جدول ۴-توضیح برچسب دادگان **qmnist**

Column	Description	Range
0	Character class	0 to 9
1	NIST HSF series	0, 1, or 4

۱ .Low-level
۲ .High-level
۳ form
۴ style
۵.Label

2	NIST writer ID	0-326 and 2100-2599
3	Digital index for this writer	0 to 149
4	NIST class code	30-39
5	Global NIST digit index	0 to 281769
6	Duplicate	0
7	Unused	0

شخص نویسنده، فرم یا استایل عدد دست‌نویس را تعیین می‌کند. ده نفر به صورت تصادفی از بین اشخاص نویسنده انتخاب‌شدند و تمام تصاویر دست‌نویس آنان با نامی به فرمت زیر در پوشه‌ای ذخیره شد:

[image number (0 to primary dataset size-1)]_[writer new id (0 to 9)]_[digit shown in the photo(0 to 9)].jpg

سپس تصاویر سیاه‌سفید و تک‌کاناله‌ی این پوشه به صورت عکس‌های رنگی سه‌کاناله‌ای در سه رنگ قرمز، آبی و سبز با فرمت نامی جدیدی ذخیره شدند. در انتها تصاویر رنگی به‌دست‌آمده از قبل، در چهار جهت چرخش صفر درجه، نود درجه، صد و هشتاد درجه و دویست و هفتاد درجه ذخیره می‌گردند. فرمت نامی تصاویر نهایی به صورت زیر است:

[image number]_[writer id]_[digit]_[color id(0 to 2)]_[rotation id(0 to 3)].jpg

برای هر یک از اشخاص در مجموعه دادگان ایجاد شده، یکی از اعداد انتخاب می‌گردد و یکی از رنگ‌ها و یا یکی از چرخش‌ها برای آن حذف می‌شوند. این دادگان حذف شده به عنوان مجموعه دادگان آزمون[۱] و باقی اعداد به عنوان مجموعه دادگان آموزش[۲] به کار می‌روند.

۳,۲. معماری شبکه

تصاویر مجموعه دادگان اعداد دست‌نویس به ابعاد 28×28 هستند؛ به همین دلیل نمی‌توان از ساختار شبکه‌ی به کار رفته در قسمت ایجاد تصاویر استفاده نمود؛ زیرا که تصاویر چهره‌ی افراد ابعادی چند برابر تصاویر اعداد دست‌نویس دارند. همچنین، برچسب‌ها به چهار بخش فرد نویسنده(استایل)، شماره‌ی موجود در تصویر، رنگ و چرخش تقسیم می‌شوند. بدین ترتیب، نیاز به تغییراتی در ساختار شبکه احساس می‌گردد.

[۱] .Test Dataset

[۲] Train Dataset

جدول ۵- بخش تولیدکننده مربوط به شبکه‌ی تصاویر اعداد دست‌نویس

Layer name	Input Feature Size	Kernel Size	Padding	Stride	Output Feature Size
Writer id FC 1	10	–	–	–	256
Leaky ReLU (negative slope=0.1)					
Writer id FC 2	256	–	–	–	512
digit FC 1	10	–	–	–	256
Leaky ReLU (negative slope=0.1)					
digit FC 2	256	–	–	–	512
color FC1	3	–	–	–	32
Leaky ReLU (negative slope=0.1)					
color FC2	32	–	–	–	64
Transform FC 1	4	–	–	–	128
Leaky ReLU (negative slope=0.1)					
Transform FC 2	128	–	–	–	256
Then	these four		outputs		being concatenated.
Leaky ReLU (negative slope=0.1)					
FC 3	1344	–	–	–	2048
Leaky ReLU (negative slope=0.1)					
FC 4	2048	–	–	–	2048
Leaky ReLU (negative slope=0.1)					
FC 5	2048	–	–	–	16384
Leaky ReLU (negative slope=0.1)					
Reshape to 256*8*8					
deConv 6	256	4	1	2	256
Leaky ReLU (negative slope=0.1)					
Conv 6	256	3	1	1	256
Leaky ReLU (negative slope=0.1)					
deConv 7	256	3	2	2	3
ReLU					

جدول ۶-بخش طبقه‌بند، مربوط به شبکه‌ی تصاویر اعداد دست‌نویس

Layer	Input Feature Size	Kernel Size	Padding	Stride	Output FeatureSize
Conv 1	3	2	0	2	128
Leaky ReLU (negative slope=0.2)					
Batch Normalization					
Conv 2	128	4	2	2	256
Leaky ReLU (negative slope=0.2)					
Batch Normalization					
Flattening Layer					
FC 3	16384	–	–	–	4096
Leaky ReLU (negative slope=0.2)					
Dropout(0.5)					
FC 4	4096	–	–	–	4096
Leaky ReLU (negative slope=0.2)					
Dropout(0.5)					
FC 5	4096	–	–	–	10
FC 3	16384	–	–	–	4096
Leaky ReLU (negative slope=0.2)					
Dropout(0.5)					
FC 4	4096	–	–	–	1024
Leaky ReLU (negative slope=0.2)					
Dropout(0.5)					
FC 5	1024	–	–	–	10
FC 3	16384	–	–	–	4096
Leaky ReLU (negative slope=0.2)					
Dropout(0.5)					
FC 4	4096	–	–	–	1024
Leaky ReLU (negative slope=0.2)					
Dropout(0.5)					
FC 5	1024	–	–	–	3

FC 3	16384	-	-	-	4096
Leaky ReLU (negative slope=0.2)					
Dropout(0.5)					
FC 4	4096	-	-	-	1024
Leaky ReLU (negative slope=0.2)					
Dropout(0.5)					
FC 5	1024	-	-	-	4

۳,۳. آموزش شبکه

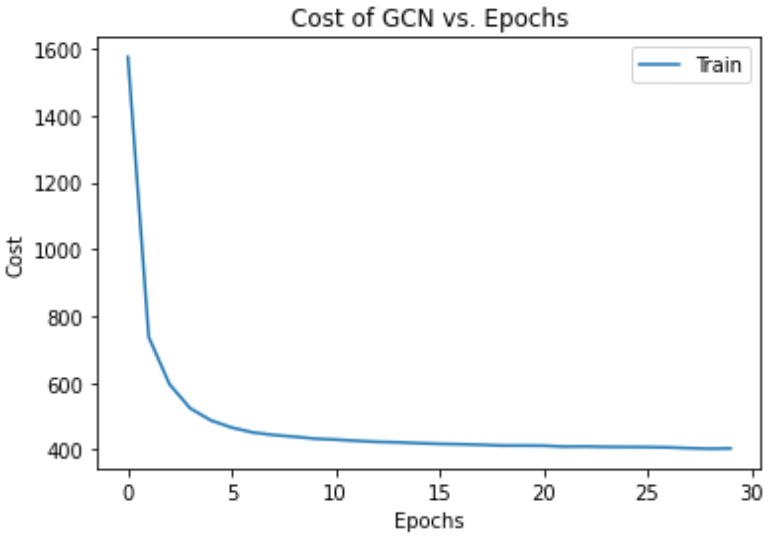
برای آموزش شبکه، از مجموعه ابرپارامترهای[۱] زیر استفاده کرده‌ایم:

جدول ۷-ابرپارامترهای استفاده‌شده در آموزش شبکه‌ی اعداد دستنویس

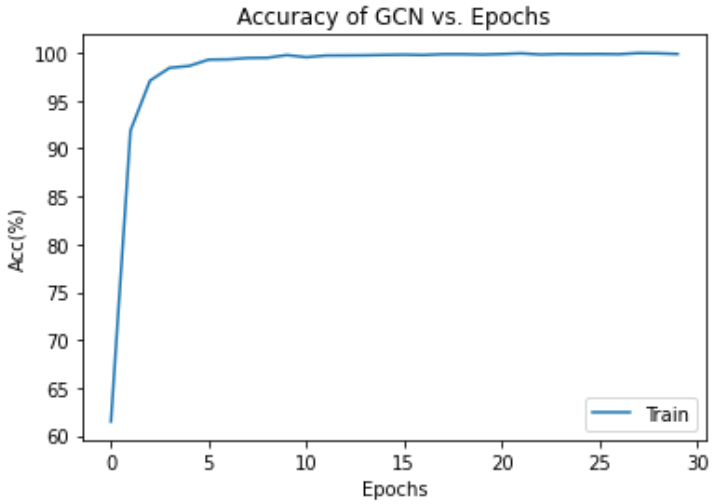
Optimizer	Adam
Learning rate	0.0002
Learning rate decay: after every 15 epochs	
β_1	0.9
β_2	0.995
ε	10^{-8}
batch size	64
epochs	30

[۱] .Hyper-parameters

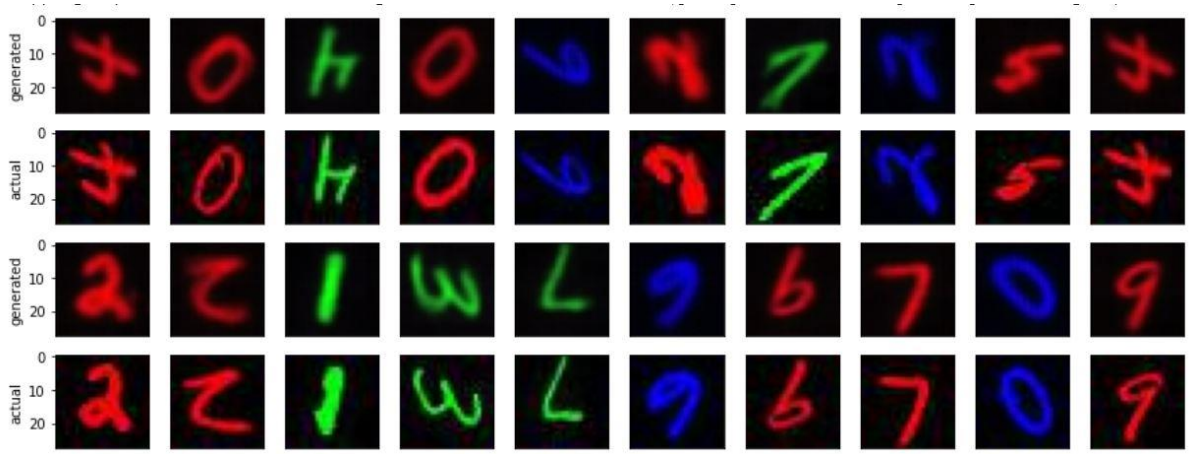
۳,۴. نتایج



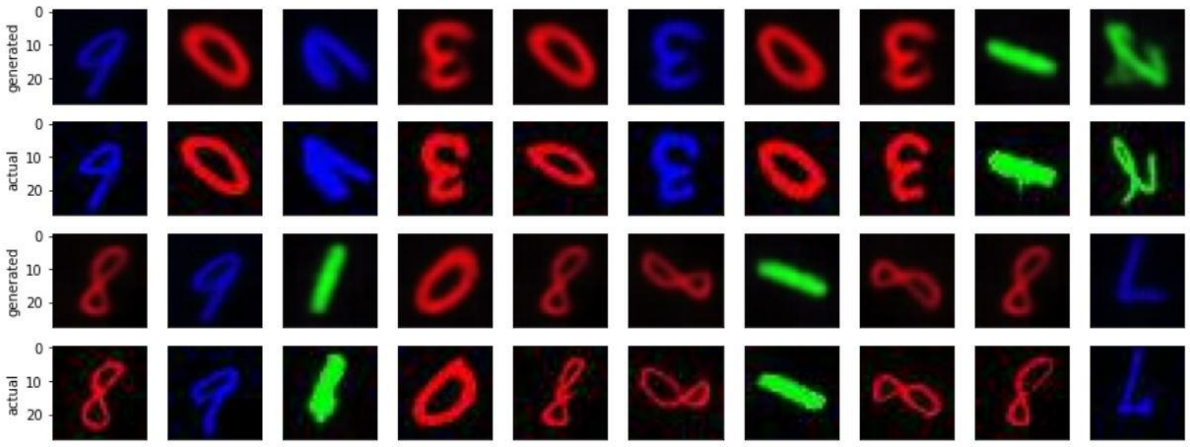
شکل ۱۰-نمودار هزینه در شبکه‌ی مربوط به اعداد دست‌نویس



شکل ۱۱-نمودار دقت در شبکه‌ی مربوط به اعداد دست‌نویس



شکل ۱۲-نمونه‌هایی از خروجی تولیدکننده شبکه در فاز آموزشی



شکل ۱۳-نمونه‌هایی از خروجی تولیدکننده شبکه در فاز آزمون

۳,۵. نتیجه‌گیری:

نتایج نشان می‌دهد که شبکه به صورت نسبی توانسته عملکرد خوبی کسب‌نماید. در تعدادی از اعداد تست، ممکن است مقداری تفاوت با استایل اصلی دیده‌شود؛ مثلاً امکان دارد قسمت‌پایینی تصویر عدد هشت در عکس اصلی، بزرگ‌تر از بخش بالایی آن باشد ولی در تصویر پیش‌بینی‌شده توسط شبکه این دو قسمت نسبتاً هم اندازه به‌نظر برسند. باید دقت کرد که پیچیدگی‌های تصاویر اعداد دست‌نویس و حتی ابعاد این تصاویر چندان نیست به همین دلیل نباید انتظار داشت که تصویر پیش‌بینی‌شده و اصلی «دقیقاً» مشابه هم باشند؛ چه بسا که یک فرد واحد در چند بار نوشتن یک عدد نمی‌تواند به صورت دقیق به یک شیوه بنویسد. بنابراین عملکرد فعلی شبکه به نظر قابل قبول است.

۴. افزودن دادگان[۱]

در این بخش در دو حالت زیر عمل طبقه بندی را روی احساس چهره افراد انجام می دهیم.

استفاده از تصاویر مجموعه دادگان KDEF

استفاده از تصاویر مجموعه دادگان KDEF به همراه تصاویر تولیدی در بخش قبلی

هدف این بخش این است که نشان دهیم استفاده از داده های تولیدی بخش قبلی می تواند دقت طبقه بندی را بالا ببرد.

۴,۱. حالت اول آموزش:

ابتدا شش نفر را انتخاب کرده و تمام تصاویر آنان را کنار می گذاریم.(در مجموع ۲۴ تصویر) تا مجموعه دادگان آزمون را تشکیل بدهند. ۶۴ نفر باقیمانده مجموعه دادگان آموزش را تشکیل می دهند (در مجموع ۲۵۶ تصویر). حال از یک شبکه عصبی عمیق برای طبقه بندی نوع احساس در تصاویر استفاده می کنیم. این شبکه عصبی دقیقا همان شبکه‌ی طبقه‌بند در بخش ابتدایی است با این تفاوت که در خروجی آن به جای سه لایه سافت مکس[۲]، فقط یک لایه سافت مکس برای پیش بینی احساس وجود دارد.

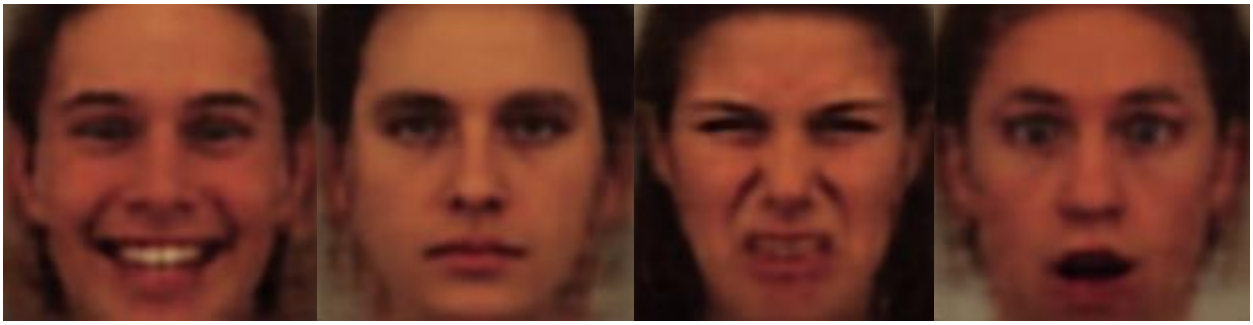
۴,۲. حالت دوم آموزش:

در این بخش علاوه بر ۲۵۶ تصویر بخش قبلی، با کمک شبکه تولیدکننده بخش قبل ۱۶۱۲۸ تصویر جدید تولید کرده و آن را به مجموعه دادگان آموزش اضافه می کنیم. شبکه تولیدکننده سه دسته ورودی دارد. ورودی اول یک بردار ۷۰ تایی برای مشخص کردن فرد است. ورودی های دوم و سوم نیز هر کدام یک بردار ۴ تایی برای مشخص کردن احساس و تبدیل هستند. برای تولید کردن نمونه جهت استفاده در طبقه بندی دو به دو افراد مختلف را که احساس و تبدیل یکسانی دارند را با هم ترکیب می کنیم. برای ترکیب کردن دو فرد، درایه‌های مربوط به آن دو فرد را در ورودی اول شبکه(بردار ۷۰ تایی) برابر ۰,۵ قرار داده و سایر درایه‌ها را صفر می گذاریم. دو ورودی دیگر شبکه نیز همان بردارهای one-hot احساس و تبدیل هستند. در شکل زیر چهار تصویر تولیدی به ازای چهار احساس متفاوت آورده شده است.

---

[۱] Data Augmentation

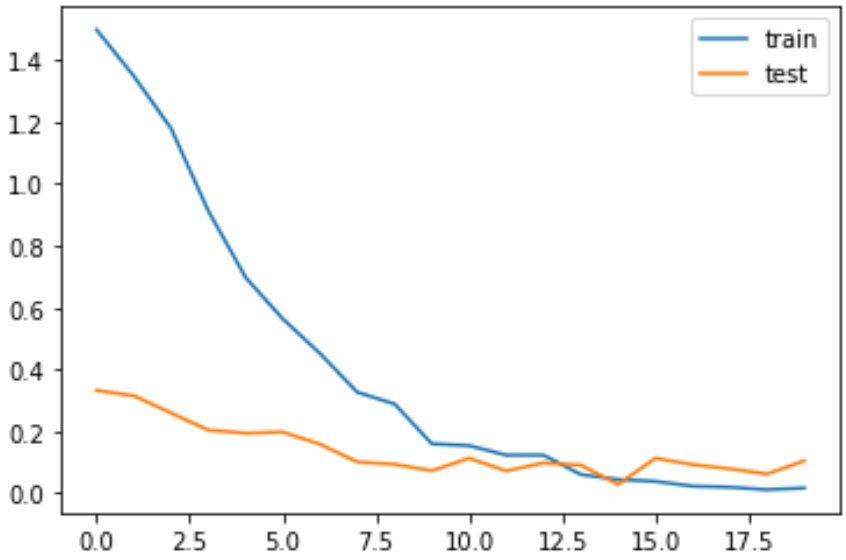
[۲] Softmax



شکل ۱۴- تصاویر افزوده، دارای احساسات متعجب، منزجر، عادی و خوشحال

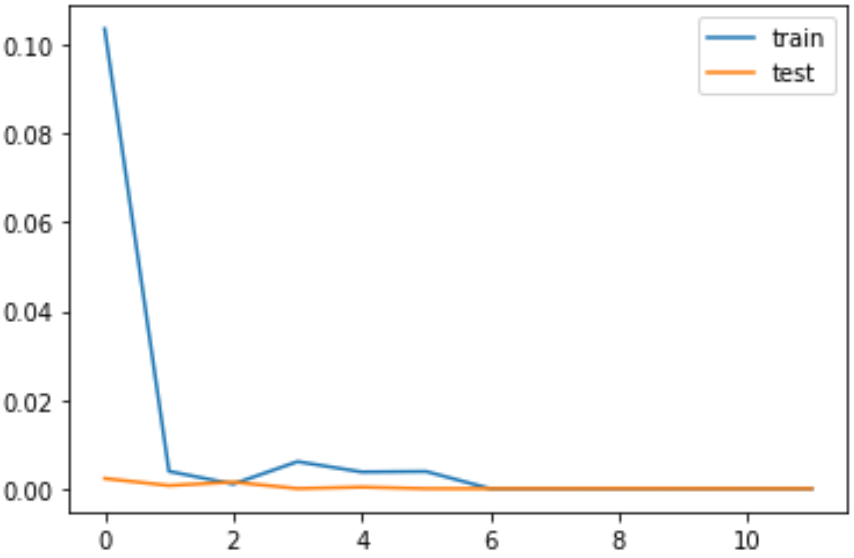
۴,۳ نتایج

در شکل زیر نمودار تابع هزینه برای داده های آموزش و آزمون حالت اول آورده شده است.



شکل ۱۵-نمودار هزینه طبقه بندی احساسات بدون افزودن داده

در شکل زیر نمودار تابع هزینه برای داده های آموزش و تست حالت دوم آورده شده است.



شکل ۱۶- نمودار هزینه طبقه بتدی احساسات پس از افزودن داده

۴,۴. نتیجه گیری

همان طور که از شکل ۱۵ پیداست، پس از ۱۴ ایپاک مدل دچار بیش‌برازش[۱] شده و لذا دقت آن بر روی مجموعه دادگان آزمون کاهش می یابد. بنابراین مقدار بهینه دقت بر روی مجموعه تست در ایپاک ۱۴ بدست می آید که برابر حدود ۹۲ درصد است. اما در شکل۱۶، با افزایش دادگان آموزش به روش افزودن دادگان مشخص می‌شود که شبکه به خوبی آموزش دیده است. در این حالت دقت بر روی مجموعه تست برابر ۱۰۰ درصد می باشد.

۱. Overfit

پیوست ۱: روند اجرای برنامه

برنامه‌ها در محیط گوگل کولب[۱] نوشته‌شدند. کدهای برنامه به صورتی نوشته‌شده‌است که نیازی به بارگذاری فایلی به صورت دستی در قسمتی نیست و می‌توان بدون نیاز به انجام عملیاتی دستی، تمام کدهارا ران[۲]کرد. تنها نکته‌ای که شاید لازم باشد به آن اشاره نمود، بارگیری مجموعه دادگان تصاویر افراد از مسیر زیر و بارگذاری آن روی گوگل درایو[۳] است:

```
!wget-P/content/drive/My\Drive/https://www.kdef.se/download/KDEF and AKDEF.zip
```

تنها یک بار ران‌کردن این قسمت برای بارگذاری داده‌ها روی گوگل درایو کافی است. این قسمت در کد اصلی موجود است و تنها عمل مورد نیاز برای ران‌کردن کدها، بارگذاری خود فایل کد روی گوگل‌کولب و ران‌کردن آن است.

[۱] Google Colab
[۲] Run
[۳] Google Drive

مراجع

[1] Xu, Q., Qin, Z., & Wan, T. (2019, March). Generative cooperative net for image generation and data augmentation. In International Symposium on Integrated Uncertainty in Knowledge Modelling and Decision Making (pp. 284-294). Springer, Cham.

[2] https://github.com/Xharlie/MultiGen

[3] https://www.programmersought.com/article/54863640507/

[4] Dosovitskiy, A., Tobias Springenberg, J., & Brox, T. (2015). Learning to generate chairs with convolutional neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1538-1546).

[5] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25, 1097-1105.