به نام خدا

دانشگاه تهران

دانشکدگان فنی

دانشکده مهندسی برق و کامپیوتر

گزارش نهایی پروژه

درس یادگیری ماشین

اعضای گروه:

حسین سیفی ۸۱۰۱۰۰۳۸۶

محمدجواد کامیاب ۸۱۰۱۰۰۴۵۷

تیر ۱۴۰۲

فهرست

استخراج فیچرها

پاکسازی داده‌ها

ابتدا تصاویر را از پوشه‌ها مورد نظر خوانده و برچسب هر کدام را یاداشت می‌کنیم این فرآیند با استفاده از کتابخانه‌ی OpenCV انجام شده.
به دلیل این که ابعاد تصاویر یکسان نبوده و نیاز است که برای تمام تصاویر به تعداد یکسانی ویژگی اختصاص یابد ابتدا تمامی تصاویر به اندازه‌ی یکسانی تبدیل می‌کنیم، مقدار انتخاب شده برای این بخش اندازه‌ی 64x64 است.
مشکل دیگری که هنگام باز کردن عکس‌ها به وجود آمد تعدادی از تصاویر غیر قابل پردازش بودند که نام این تصاویر به شرح زیر است:

- 810199515_real_none_jungle_1.jpeg
- 810199515_real_none_jungle_10.jpeg
- 810197636_real_none_sea_4.jpeg

این سه تصویر از مجموعه‌ی داده جدا شدند و ادامه‌ی فرآیند استخراج ویژگی برای سایر تصاویر انجام شد.

ویژگی‌های آماری

به عنوان ویژگی‌های پایه، می‌توان ویژگی‌های آماری تصاویر را استخراج کرد و نتایج را مورد بررسی قرار داد. این ویژگی‌های آماری شامل میانگین، حداکثر، حداقل، انحراف معیار و چولگی[1] هر یک از ابعاد رنگی تصاویر (قرمز، سبز و آبی) می‌شوند. با این روش برای هر تصویر یک فضای ویژگی ۱۵ بعدی ایجاد شده است که با بررسی آن‌ها مشخص شد که طبقه‌بندهای متفاوت با این ویژگی‌ها دقتی نه چندان قابل توجه ارائه می‌دهند و تنها کمی بهتر از طبقه‌بند تصادفی عمل می‌کنند.

HOG feature extraction

هیستوگرام گرادیان های جهت دار (HOG) یک تکنیک توصیفگر ویژگی محبوب در بینایی کامپیوتری و پردازش تصویر است. توزیع جهت های لبه را در یک شی تجزیه و تحلیل می کند تا شکل و ظاهر آن را توصیف کند. روش HOG شامل محاسبه مقدار گرادیان و جهت گیری برای هر پیکسل در یک تصویر و سپس تقسیم تصویر به سلول های کوچک است.
برخی از جنبه های مهم HOG نگاه کنیم که آن را از سایر توصیفگرهای ویژگی متفاوت می کند:
توصیفگر HOG بر ساختار یا شکل یک شی تمرکز می کند. برخلاف ویژگی های لبه که فقط تشخیص می‌دهد که آیا پیکسل یک لبه است یا نه HOG .قادر است جهت لبه را نیز ارائه دهد. این کار با استخراج گرادیان و جهت (یا می توان گفت بزرگی و جهت) لبه ها انجام می شود.
علاوه بر این، این جهت گیری ها در بخش های "محلی" محاسبه می شوند. این بدان معنی است که تصویر کامل به مناطق کوچکتر تقسیم می شود و برای هر منطقه، گرادیان ها و جهت گیری محاسبه می شود.
در نهایت HOG یک هیستوگرام برای هر یک از این مناطق به طور جداگانه ایجاد می کند. هیستوگرام ها با استفاده از گرادیان ها و جهت گیری های مقادیر پیکسل ایجاد می شوند، از این رو "هیستوگرام گرادیان های جهت دار" نامیده می شود.

در اینجا توضیح گام به گام فرآیند استخراج ویژگی HOG آمده است:
پیش پردازش: تصویر ورودی برای افزایش کیفیت و مناسب تر کردن آن برای استخراج ویژگی پیش پردازش شده است. مراحل پیش پردازش معمولی شامل تغییر اندازه تصویر به اندازه ثابت، تبدیل آن به مقیاس خاکستری و اعمال عادی سازی کنتراست است.
محاسبه گرادیان: گرادیان های تصویر برای ثبت تغییرات شدت محلی محاسبه می شوند. تصویر با دو فیلتر، معمولاً فیلترهای سوبل، در جهت افقی و عمودی در هم می پیچد. سپس گرادیان ها به عنوان بزرگی و جهت گیری مقادیر گرادیان افقی و عمودی حاصل محاسبه می شوند.
تقسیم سلولی: تصویر به سلول های کوچک، معمولاً ۸ x 8 پیکسل تقسیم می شود. این تقسیم بندی امکان ثبت الگوهای محلی درون تصویر را فراهم می کند و در عین حال تاثیر نویز و تغییرات کوچک را کاهش می دهد.
محاسبه هیستوگرام: برای هر سلول، یک هیستوگرام جهت گیری گرادیان ساخته می شود. جهت هر گرادیان به تعداد گسسته ای از دسته‌ها (به عنوان مثال، ۹ دسته از ۰ تا ۱۸۰ درجه) کوانتیزه می شود. بزرگی گرادیان به عنوان وزن برای رأی گیری در هیستوگرام استفاده می شود. هر binآرای وزن شده را از گرادیان های داخل سلول جمع می کند.

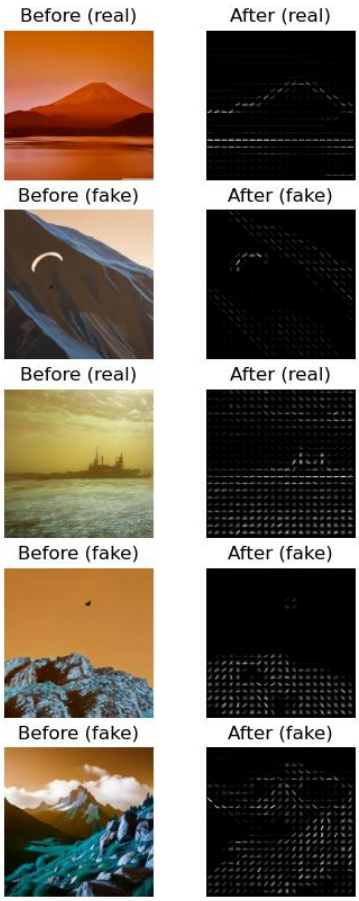
---

[1] Skewness

نرمال سازی بلوک: برای مدیریت تغییرات نور و بهبود استحکام، سلول های مجاور با هم در بلوک ها گروه بندی می شوند. نرمال سازی بلوک با اعمال یک تابع نرمال‌ساز، مانندL2-normalization ، به مقادیر هیستوگرام در هر بلوک انجام می شود. این فرآیند بلوک را نرمال می کند و آن را نسبت به تغییرات شدت موضعی کمتر حساس می کند.
بردار ویژگی: بردار ویژگی نهایی HOG با الحاق مقادیر هیستوگرام نرمال شده از تمام بلوک های تصویر به دست می آید. این بردار ویژگی توزیع جهت های گرادیان محلی را ثبت می کند و نمایش فشرده ای از تصویر را ارائه می دهد.
هنگامی که ویژگی های HOG استخراج می شوند، می توانند به عنوان ورودی برای الگوریتم های مختلف یادگیری ماشین، مانند ماشین های بردار پشتیبان (SVM) یا شبکه های عصبی، برای طبقه بندی استفاده شوند. این الگوریتم ها یاد می گیرند که بین تصاویر واقعی و جعلی بر اساس الگوهای ثبت شده توسط ویژگی های HOG تمایز قائل شوند. ویژگی های HOG اطلاعات مربوط به شیب های محلی و جهت گیری در تصاویر را می گیرد. در زمینه طبقه‌بندی تصاویر واقعی در مقابل جعلی، این ویژگی‌ها می‌توانند به ثبت تفاوت‌ها در بافت، شکل و سایر نشانه‌های بصری کمک کنند که ممکن است تصاویر واقعی را از تصاویر جعلی تولید شده توسط هوش مصنوعی متمایز کند.
ویژگی های HOG ممکن است با ثبت تفاوت در الگوهای بافتی، لبه ها و اطلاعات شکل، به تشخیص تصاویر واقعی از تصاویر جعلی تولید شده توسط هوش مصنوعی کمک کند. در حالی که ویژگی های دقیق تصاویر جعلی واقعی و تولید شده توسط هوش مصنوعی می تواند متفاوت باشد، برخی نشانه های بصری بالقوه وجود دارد که ویژگی های HOG می توانند ثبت کنند:

بافت: تصاویر واقعی گرفته شده توسط انسان اغلب بافت ها و تغییرات طبیعی را نشان می دهند، در حالی که تصاویر جعلی تولید شده توسط هوش مصنوعی ممکن است بافت های صاف یا یکنواخت تری داشته باشند. ویژگی‌های HOG می‌توانند تغییرات بافت را با تجزیه و تحلیل توزیع محلی جهت‌گیری‌های گرادیان، که می‌تواند بین تصاویر واقعی و جعلی تمایز قائل شود، ثبت کند.
شکل و ساختار: تصاویر واقعی معمولاً اشکال و ساختارهای طبیعی و ثابتی را نشان می دهند، در حالی که تصاویر جعلی تولید شده توسط هوش مصنوعی ممکن است دارای ناهنجاری یا ناهماهنگی در شکل خود باشند. ویژگی‌های HOG می‌توانند اطلاعاتی در مورد لبه‌ها، خطوط و شیب‌های محلی به دست آورند و به مدل اجازه می‌دهند تا تغییرات شکل و ساختار را شناسایی کند.
جزئیات با فرکانس بالا: تصاویر واقعی ممکن است حاوی جزئیات با فرکانس بالا باشند که تکرار آن در تصاویر جعلی تولید شده توسط هوش مصنوعی چالش برانگیز است. ویژگی‌های HOG می‌توانند این جزئیات دقیق را از طریق تجزیه و تحلیل گرادیان به تصویر بکشند و مدل را قادر می‌سازند تا بین دو دسته بر اساس وجود یا عدم وجود چنین جزئیاتی تمایز قائل شود.

تشخیص مصنوع: تصاویر جعلی تولید شده توسط هوش مصنوعی ممکن است مصنوعات یا الگوهای خاصی را برای فرآیند تولید نشان دهند. ویژگی‌های HOG می‌توانند این مصنوعات را به‌عنوان انحراف از گرادیان‌ها و جهت‌گیری‌های مورد انتظار، به‌طور بالقوه ارائه اطلاعات متمایزکننده برای طبقه‌بندی، ثبت کنند.
برای پیاده‌سازی این ویژگی و ویژگی بعدی از کتاب‌خانه‌ی skimage استفاده شده است.
هنچنین نمونه‌ای از خروجی این فیلتر به صورت رو به رو است.



LBP feature extraction

. (Local Binary Patterns) LBPیک تکنیک استخراج ویژگی محبوب است که در پردازش تصویر و بینایی رایانه ای برای کارهایی مانند تجزیه و تحلیل بافت و طبقه بندی تصویر استفاده می شود. در ثبت الگوهای بافت محلی در یک تصویر موثر است. عملگر الگوی باینری محلی یک کد باینری تولید می‌کند در حالی که پیکسل همسایه و واحد خاکستری وصله مرکزی آن را مقایسه می‌کند. اگر پیکسل همسایه کوچکتر از مقدار مرکزی باشد، این عملگر ۰ را اختصاص می دهد. در غیر این صورت، یک مقدار واحد اختصاص می دهد LBP .توصیفگر بسیار قدرتمندی است که تمام لبه های ممکن در تصویر را تشخیص می دهد.

فرآیند استخراج ویژگی LBP به‌صورت زیر می‌باشد:

پیش پردازش: تصاویر ورودی را با تبدیل آن‌ها به مقیاس خاکستری در صورتی که رنگی هستند، از قبل بایستی پردازش کرد LBP .بر روی تصاویر خاکستری عمل می کند. تصاویر باید دارای اندازه ثابت برای استخراج ویژگی باشند.

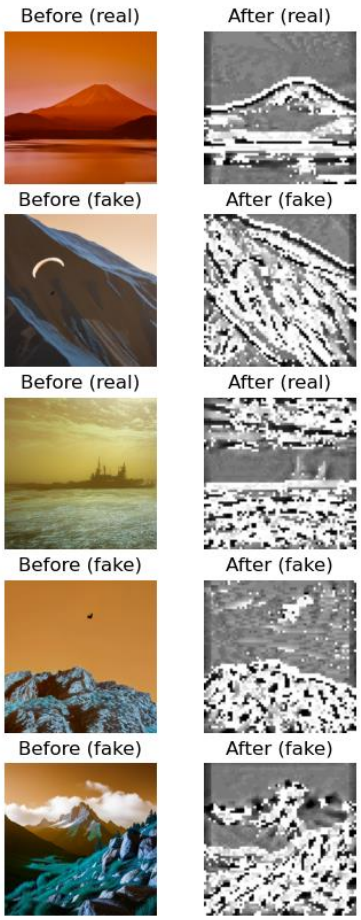
تعریف محله تصویر: برای هر پیکسل در تصویر، یک محله در اطراف آن تعریف می‌کند. محله معمولاً یک ناحیه دایره ای است که در مرکز پیکسل قرار دارد.

محاسبه :LBP برای هر پیکسل، مقدار شدت آن را با شدت پیکسل های اطراف در همسایگی مقایسه می‌کند. اگر شدت پیکسل اطراف بیشتر یا مساوی با شدت پیکسل مرکزی باشد، یک مقدار باینری ۱ اختصاص می‌دهد. در غیر این صورت، مقدار باینری ۰ را تعیین می‌کند. این فرآیند بر اساس مقایسه محلی شدت ها، یک الگوی باینری برای هر پیکسل ایجاد می‌کند.

محاسبه هیستوگرام: با شمارش وقوع الگوهای باینری مختلف در تصویر، یک هیستوگرام می‌سازد. هر الگوی باینری منحصر به فرد یک bin در نظر گرفته می شود و هیستوگرام فرکانس هر الگو را جمع می‌کند.

بردار ویژگی: بردار نهایی ویژگی LBP با الحاق مقادیر هیستوگرام به دست می‌آید. طول بردار ویژگی به تعداد الگوهای باینری منحصر به فرد در تصویر بستگی دارد. هر bin در هیستوگرام نشان دهنده بسامد یک الگوی خاص است که در تصویر رخ می‌دهد.

دلایل انتخاب:LBP



تمایز بافت LBP :به دلیل توانایی خود در گرفتن و نمایش موثر اطلاعات بافت شناخته شده است. تصاویر واقعی گرفته شده توسط انسان و تصاویر جعلی تولید شده توسط هوش مصنوعی اغلب ویژگی های بافتی متمایزی را نشان می دهند LBP .می‌تواند این الگوهای بافت را با تجزیه و تحلیل همسایگی محلی پیکسل‌ها ثبت کند و برای تشخیص تصاویر واقعی و جعلی بر اساس ویژگی‌های بافتی منحصربه‌فردشان مناسب باشد.

تجزیه و تحلیل موضعی LBP :بر روی اطلاعات محلی یک تصویر تمرکز می کند و رابطه بین پیکسل مرکزی و همسایگان اطراف آن را تجزیه و تحلیل می کند. این تجزیه و تحلیل موضعی به LBP اجازه می دهد تا جزئیات دقیق و الگوهای بافت محلی را ثبت کند، که می تواند ویژگی های متمایز کننده ای برای تمایز بین تصاویر واقعی و جعلی باشد.

تغییر ناپذیر نسبت به تغییرات روشنایی LBP: نسبت به تغییرات نور در تصاویر قوی است. به جای تکیه بر مقادیر شدت مطلق، شدت پیکسل ها را با همسایگان خود مقایسه می کند. این ویژگی باعث می‌شود ویژگی‌های LBP نسبت به تغییرات در شرایط نوری که می‌توانند هم در تصاویر واقعی و هم در تصاویر جعلی وجود داشته باشند، حساسیت کمتری داشته باشند. در نتیجه، LBPمی تواند به طور موثر الگوهای بافت ذاتی را با وجود تغییرات روشنایی به تصویر بکشد.

تفسیرپذیری: ویژگی‌های LBP تفسیر روشنی دارند زیرا فرکانس‌های وقوع الگوهای بافت خاص را نشان می‌دهند. این تفسیرپذیری می‌تواند بینش‌هایی را در مورد ویژگی‌های بافتی متمایز بین تصاویر واقعی و جعلی ارائه دهد، به درک فرآیند طبقه‌بندی کمک می‌کند و به طور بالقوه به شناسایی عوامل کلیدی که این دو کلاس را متمایز می‌کند کمک می‌کند.

نمونه‌ای این فیلتر بر روی دیتاست داده شده به صورت روبه‌رو است:

طبقه‌بندی

در این بخش از پروژه به طبقه‌بندی داده‌ها در دو بخش می‌پردازیم. پیش از انجام طبقه‌بندی بر روی وظیفه مورد نظر، نیاز به انجام پیش‌پردازش وجود دارد تا داده‌ها را برای انجام عمل آموزش و تست مدل‌ها آماده کند. سپس اولین دسته مدل‌های طبقه‌بند دو کلاسه با استفاده از ویژگی‌های آماده‌ی موجود ایجاد می‌شوند که داده‌ها را به دو دسته واقعی و غیرواقعی طبقه‌بندی می‌کنند. در نهایت با استفاده از ویژگی‌های استخراج شده توسط تیم خود، داده‌ها طبقه‌بندی می‌شوند و نتایج مدل‌های هر دو دسته با یکدیگر مقایسه می‌شوند.

پیش پردازش

در فایل نام تصاویر متناظر با ویژگی‌های آماده، دو نام Desktop.ini وجود دارد که مشخص نیست که به طبقه داده‌های واقعی تعلق دارند یا غیرواقعی، بنابراین این دو نام به همراه ویژگی‌های با اندیس مشابه در فایل ویژگی‌ها از داده‌ها حذف می‌شوند. همچنین داده‌ها به کمک کتابخانه Scikit-learn با نسبت ۷۰ درصد به ۳۰ درصد به دو بخش آموزش و تست تقسیم شده‌اند.

طبقه‌بندی با استفاده از ویژگی‌های آماده

با توجه به عدم تفسیرپذیری ویژگی‌هایی که در اختیار ما قرار گرفته است و تعداد ابعاد بالای داده‌ها که تحلیل آن‌ها را به منظور انتخاب مدل مناسب برای داده‌های در دسترس مشکل می‌کند، تنها راه انتخاب مدل مناسب، آزمودن مدل‌های در دسترس و بررسی معیارهای ارزیابی برای انتخاب مدلی با بهترین عملکرد است. بنابراین در این بخش ۶ مدل یادگیری ماشین با پارامترهای مناسب بر روی داده‌های موجود آموزش دیده‌اند. این مدل‌ها شامل ماشین بردار پشتیبان[۲]، درخت تصمیم، شبکه عصبی MLP، Naïve Bayes، Logistic Regression و K نزدیک‌ترین همسایه[۳] هستند. برای هر مدل معیارهای ارزیابی محاسبه شده‌اند، ماتریس آشفتگی رسم شده است و نمونه‌هایی از بخش تست که در دسته اشتباه قرار گرفته‌اند به نمایش درآمده‌اند. با توجه به اینکه تعداد نمونه‌های موجود در دو کلاس برابر است، نیازی به محاسبه هر معیار ارزیابی به دو صورت Macro و Micro دیده نمی‌شود و برای هر معیار، مقادیر مربوط به کلاس‌ها به همراه مقدار مربوط به کل داده‌ها اعلام می‌شود.

ماشین بردار پشتیبان

ماشین بردار پشتیبان یک الگوریتم بانظارت یادگیری ماشین است که برای مسائل طبقه‌بندی و رگرسیون قابل استفاده است. در الگوریتم SVM، هر نمونه داده را به کمک تابع کرنل، به تعداد ابعاد بالاتری می‌برد و سپس با ترسیم یک خط راست، داده‌های کلاس‌های متفاوت را طبقه‌بندی می‌کند. در مدل استفاده شده برای ویژگی‌های موجود و طبقه‌بندی تصاویر واقعی و غیرواقعی، تابع کرنل RBF[۴] انتخاب شده است چراکه بهترین عملکرد را از خود نشان می‌دهد. این مدل بر روی داده‌های مربوطه آموزش دید و مقادیر معیارهای ارزیابی آن بر روی داده‌های آموزش و تست به شرح زیر است:

	Precision	Recall	F1-Score	Accuracy
آموزش	۰.۹۹۷۱	۰.۹۹۷۱	۰.۹۹۷۱	۰.۹۹۷۱
تست	۰.۹۹۶۲	۰.۹۹۶۰	۰.۹۹۶۱	۰.۹۹۶۱

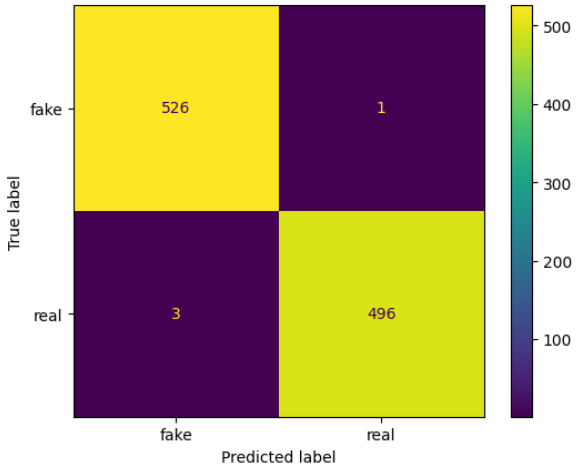
همچنین ماتریس آشفتگی[۵] برای داده‌های تست به شکل زیر خواهد بود:

[2] Support Vector Machine(SVM)

[3] K-Nearest-Neighbor(KNN)

[4] Radial Basis Function

[5] Confusion Matrix



همانگونه که مشخص این مدل دقت بسیار خوبی بر روی دادگان آموزش از خود نشان داده است که می‌تواند به معنی به دست آوردن ابرصفحه[۶] مناسب برای تفکیک داده‌ها دو کلاس باشد. همچنین اندک خطایی در تفکیک داده‌های آموزش وجود دارد که به معنی استفاده از روش‌های Soft margin برای به دست آوردن این ابرصفحه است. در این روش ابرصفحه به نوعی به دست می‌آید که تعادلی بین داده‌هایی که به اشتباه طبقه‌بندی می‌شوند و اندازه حاشیه[۷] ابرصفحه به وجود آید. همچنین دقت مدل بر روی دادگان تست نیز بسیار خوب است و اندکی از دقت داده‌های آموزش کمتر است. این کاهش ناچیز دقت به دلیل وجود خطای ذاتی در مدل‌های یادگیری ماشین طبیعی است و نمی‌توان به عنوان بیش برازش[۸] تفسیر شود. در ادامه دو تصویر از بخش تست که به اشتباه طبقه‌بندی شده‌اند به همراه برچسب خود، به نمایش درآمده‌اند:

Real

Fake

برای تشخیص واقعی یا غیرواقعی بودن دو تصویر فوق بدون استفاده از مدل‌های یادگیری ماشین نیز با چالش روبرو هستیم و برچسب آن‌ها به درستی قابل تشخیص نیست. این چالش برای مدل SVM آموزش دیده نیز به وجود آمده است و برچشب اشتباهی برای این دو تصویر انتخاب کرده است. بنابراین خطا در انتخاب برچسب برای این تصاویر را نمی‌توان به عنوان خطای مدل در نظر گرفت و می‌تواند به دلیل انتخاب برچسب اشتباه برای نمونه‌ها، پیروی نمونه‌های فوق از توزیع متفاوتی نسبت به سایر نمونه‌های تست و آموزش، و سایر مشکلات مشابه به وجود آمده باشد.

درخت تصمیم

درخت تصمیم یک الگوریتم حریصانه یادگیری ماشین است که در هر مرحله تلاش می‌کند با انتخاب بهترین ویژگی ممکن بی‌نظمی را در هر شاخه درخت کاهش دهد تا جایی که داده‌ها از یکدیگر قابل تفکیک باشند. اگرچه به نظر می‌رسد این الگوریتم بر روی داده‌هایی با تعداد ابعاد

---

[6] Hyperplane

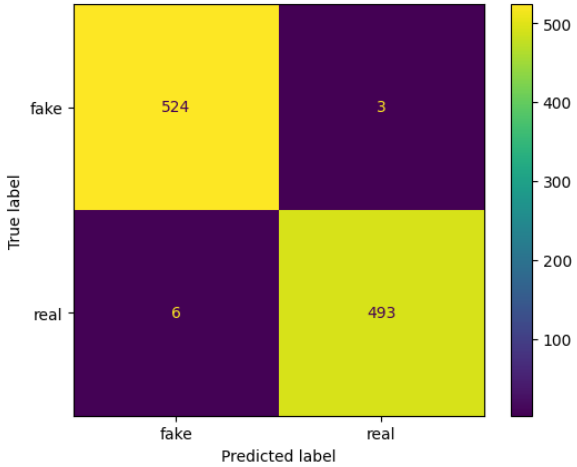
[7] Margin

[8] Overfitting

بالا عملکرد مناسبی را از خود نشان نخواهد داد و باعث ایجاد بیش‌برازش خواهد شد، نتایج این الگوریتم حتی بدون در نظر گرفتن پارامتری مثل حداکثر عمق درخت، رضایت‌بخش بود. این نتایج در جدول زیر قابل مشاهده است:

	Precision	Recall	F1-Score	Accuracy
آموزش	۱	۱	۱	۱
تست	۰.۹۹۱۳	۰.۹۹۱۱	۰.۹۹۱۲	۰.۹۹۱۲

و ماتریس آشفتگی درخت تصمیم به شکل زیر است:



مطابق انتظار دقت این مدل بر روی داده‌های آموزش برابر با ۱ است ولی به دلیل پیروی تقریبی داده‌های تست از توزیع داده‌های آموزش، بیش‌برازش اتفاق نیافتاده است و مدل درخت تصمیم بر روی داده‌های تست نیز عملکرد قابل قبول، اگرچه کمی ضعیف‌تر از خود نشان داده است. اگرچه ممکن با انتخاب حداکثر عمقی مناسب برای این درخت، فاصله بین دقت داده‌های آموزش و تست از آنچه که هست، کمتر شود. چند نمونه از داده‌های تست که به اشتباه طبقه‌بندی شده‌اند در ادامه قابل مشاهده است:

Real

Fake

دو تصویری که مدل SVM اشتباه طبقه‌بندی کرده بود، توسط مدل درخت تصمیم نیز به اشتباه طبقه‌بندی شده‌اند و این نشان دهنده عدم تطابق این تصاویر با سایر نمونه‌های تست و آموزش است. همچین تصاویری که باعث ایجاد خطا در مدل‌ها می‌شوند را می‌توان در مرحله پیش‌پردازش از بین داده‌ها حذف کرد چرا که در صورت وجود این تصاویر در مجموعه آموزش می‌توانند باعث به وجود آمدن یک مدل طبقه‌بندی شوند که به داده‌های دور افتاده[9] و پرت نیز حساس است.

شبکه عصبی MLP

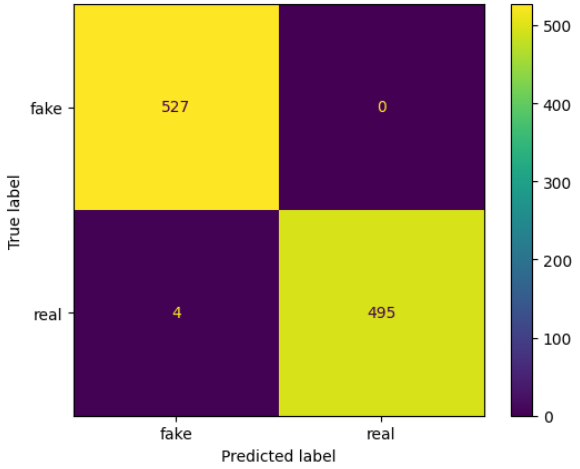
مدل MLP یا Multi-Layer-Perceptron یک شبکه عصبی مصنوعی از نوع Fully connected است. این شبکه عصبی حداقل دارای ۳ لایه ورودی، نهان و خروجی است. لایه ورودی به اندازه تعداد ویژگی‌های هر نمونه، لایه خروجی به اندازه تعداد کلاس‌های مجموعه داده و

[9] Outliers

لایه‌های نهان به تعداد دلخواه نورون دارند. در شبکه عصبی آموزش دیده بر روی دادگان موجود، از دو لایه نهان با اندازه ۶۴۰ و ۳۲۰ استفاده شده است. این اندازه لایه‌ها با روشی تجربی انتخاب شده‌اند. این روش بیان می‌کند تعداد نورون‌ها در هر لایه برابر با نصف تعداد نورون‌های لایه قبل باشد اما در شبکه تعریف شده این روش بررسی شد و نتیجه‌گیری شد که تعداد لایه‌های بیشتر از ۲، با تعداد نورون‌های ذکر شده باعث افزایش دقت شبکه نمی‌شوند و تنها هزینه محاسباتی و استفاده بیشتر از منابع را برای ما به همراه دارند. مقادیر معیارهای ارزیابی شبکه آموزش دیده بر روی داده‌های آموزش و تست به شرح زیر است:

	Precision	Recall	F1-Score	Accuracy
آموزش	۱	۱	۱	۱
تست	۰.۹۹۶۲	۰.۹۹۶۰	۰.۹۹۶۱	۰.۹۹۶۱

ماتریس آشفتگی برای مدل MLP به شکل زیر به دست آمده است:



این مدل نیز همانند درخت تصمیم بر روی داده‌های آموزش بهترین عملکرد ممکن را از خود نشان می‌دهد و حتی یک داده را به اشتباه طبقه‌بندی نمی‌کند. اما بر روی دادگان تست اندک خطایی وجود دارد که همانند مدل‌های قبل به عنوان خطای ذاتی مدل یا عدم تطابق توزیع دادگان تست با دادگان آموزش تفسیر می‌شود. دو نمونه از تصاویر مجموعه تست که به اشتباه طبقه‌بندی شده‌اند به شکل زیر هستند:

Real

Real

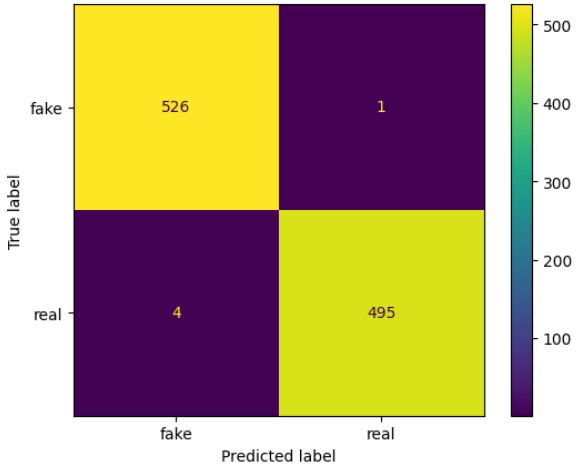
بر خلاف تصاویری که طبقه‌بندهای قبلی به اشتباه برچسب زده بودند، این دو تصویر به راحتی توسط انسان قابل تشخیص هستند و برخلاف عملکرد مناسب مدل MLP بر روی دادگان آموزش، بر روی تعدادی از دادگان بخش تست از جمله دو تصویر فوق را دچار اشتباه شده است.

Naïve Bayes

در مدل‌های Naïve Bayes با فرض استقلال ویژگی‌های داده‌ها از یکدیگر، توزیع احتمالات کلاس‌ها تخمین زده می‌شود و سپس به کمک احتمالات ایجاد شده، احتمال تعلق نمونه‌ها به هر کلاس محاسبه می‌شود. سپس نمونه برچسبی را دریافت می‌کند که دارای بیشترین احتمال است. فرض استقلال ویژگی‌ها یک فرض بزرگ است و در صورتی که ویژگی‌های یک مجموعه داده به یکدیگر وابسته باشند، این مدل‌ها عملکرد خوبی از خود نشان نمی‌دهند. در جدول زیر مقادیر معیارهای ارزیابی این مدل بر روی داده‌های آموزش و تست ذکر شده است:

	Precision	Recall	F1-Score	Accuracy
آموزش	۰.۹۹۶۶	۰.۹۹۶۷	۰.۹۹۶۷	۰.۹۹۶۷
تست	۰.۹۹۵۲	۰.۹۹۵۰	۰.۹۹۵۱	۰.۹۹۵۱

ماتریس آشفتگی به دست آمده برای این مدل بر روی داده‌های تست نیز به شکل زیر است:



همانطور که مشخص است این مدل عملکرد تقریبا یکسان و بسیار خوبی بر روی هر دو بخش آموزش و تست از خود نشان داده است. طبق آنچه گفته شد، این عملکرد خوب مدل Naïve Bayes تنها می‌تواند به دلیل استقلال ویژگی‌های این مجموعه داده باشد در غیر اینصورت امکان مشاهده همچین عملکردی وجود نداشت. در ادامه دو نمونه از تصاویری که توسط این مدل به اشتباه طبقه‌بندی شده‌اند به نمایش درآمده است:

Fake

Real

این دو تصویر نقطه مشترک اشتباهات اکثر مدل‌های طبقه‌بندی فوق در طبقه‌بندی تصاویر واقعی و غیرواقعی هستند که دلایل احتمالی آن در بخش‌های قبل مورد بررسی قرار گرفت.

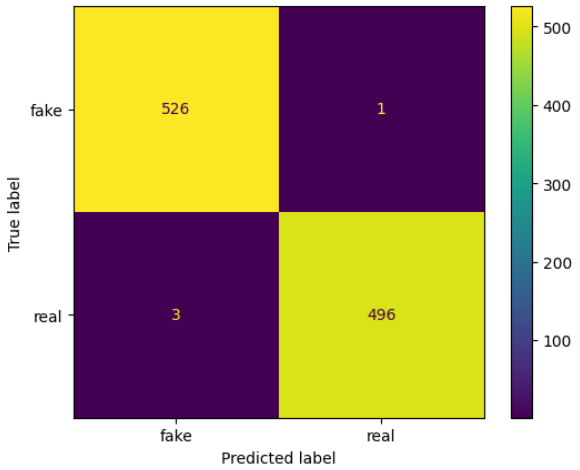
Logistic Regression

مدل Logistic Regression با استفاده از روش گرادیان کاهشی[10] تلاش می‌کند تا بهترین مقادیر وزن‌ها را برای ایجاد جدا کننده دو کلاس مجموعه داده به دست آورد. این روش مشکلاتی از جمله متوقف شدن در نقطه بهینه محلی را دارد اما همچنان یکی از بهترین مدل‌های یادگیری ماشین است. در جدول زیر مقادیر معیارهای ارزیابی این مدل بر روی داده‌های آموزش و تست قابل مشاهده است:

[10] Gradient Descent

	Precision	Recall	F1-Score	Accuracy
آموزش	۱	۱	۱	۱
تست	۰.۹۹۶۲	۰.۹۹۶۰	۰.۹۹۶۱	۰.۹۹۶۱

و ماتریس آشفتگی برای داده‌های تست به شکل زیر است:



عملکرد این مدل نیز همانند تعدادی دیگر از مدل‌های آموزش دیده مانند شبکه عصبی MLP است. این عملکرد نیز قابل است اما مقدار ۱ برای تمامی معیارهای ارزیابی روی داده‌های آموزش و مقادیری کمتر برای داده‌های تست، نشان دهنده این است که مدل Logistic Regression پتانسیل بیش برازش را دارا است. چند نمونه از تصاویری که به اشتباه طبقه‌بندی شده‌اند به شکل زیر هستند:

Fake

Real

و باز هم تصاویر مشترکی که به اشتباه طبقه‌بندی شده‌اند، بین این مدل و مدل‌های قبلی وجود دارد.

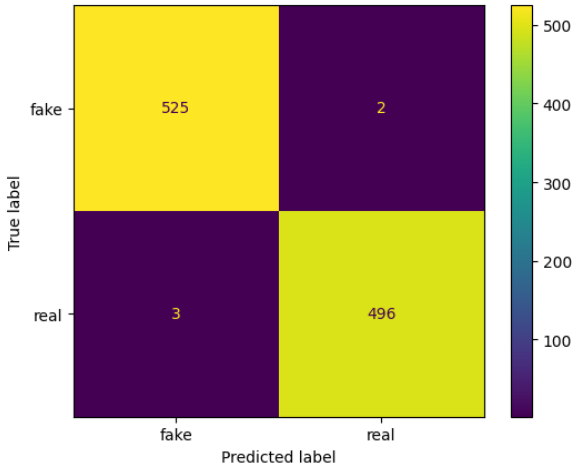
K-نزدیکترین-همسایه

این مدل از دسته مدل‌های یادگیرنده تنبل[11] یادگیری ماشین است. این مدل‌ها به یادگیری از نمونه‌های آموزش نمی‌پردازند و تنها روشی بهینه برای ذخیره‌سازی این داده‌ها دارند. سپس در مرحله تست به انجام پردازش می‌پردازند. این پردازش مرحله تست برای مدل‌های K-نزدیکترین همسایه، محاسبه فاصله تا سایر نمونه‌ها و سپس انتخاب برچسب بر اساس اکثریت بین K همسایه نزدیک‌تر است. مقدار معیارهای ارزیابی برای این مدل با K برابر با ۵ به شرح زیر است:

	Precision	Recall	F1-Score	Accuracy
آموزش	۰.۹۹۷۹	۰.۹۹۷۹	۰.۹۹۷۹	۰.۹۹۷۹
تست	۰.۹۹۵۲	۰.۹۹۵۱	۰.۹۹۵۱	۰.۹۹۵۱

[11] Lazy Learner

و ماتریس آشفتگی به شکل زیر برای داده‌های تست به دست می‌آید:



همچنین نمونه‌های زیر به اشتباه دسته‌بندی شده‌اند:

Real

Fake

مانند مدل‌های قبل، این مدل نیز با نمونه‌های یکسانی دچار چالش شده است.

مقایسه مدل‌ها

به طور مفصل برای هر بخش توضیح داده شد که عملکرد هر مدل بر روی دادگان موجود چگونه است. معیارهای ارزیابی محاسبه شده دقت، صحت، فراخوانی و امتیاز F1 برای ۶ مدل ذکر شده بسیار مشابه با یکدیگر به دست آمد. بعضی مدل‌ها مانند MLP، درخت تصمیم و Logistic Regression نشان دادند که در شرایطی خاص و با مجموعه داده‌هایی دیگر، دارای امکان بیش‌برازش هستند اما دیگر مدل‌ها بر روی مجموعه آموزش و تست دقت مشابهی را به نمایش گذاشته‌اند که حتی کوچکترین احتمال بیش‌برازش نیز دیده نمی‌شود. همچنین تمامی این مدل‌ها پیچیدگی نسبتا یکسانی دارند که باعث می‌شود مقدار خطای بایاس و واریانس همه در وضعیت مشابهی باشد و در هیچ‌یک از مدل‌ها تعادل بین این دو خطا به هم نخورده است. همانطور که به نمایش درآمد تمامی مدل‌ها برای تعیین برچسب تصاویر یکسانی به چالش برخوردند که به دلایلی مانند عدم پیروی تصاویر مذکور از توزیع سایر تصاویر و به خصوص تصاویر آموزش، عدم انجام صحیح عمل برچسب‌زنی برای تصاویری که به صورت مکرر مدل‌ها را به خطا وادار می‌کند و ... ممکن است اتفاق بیافتد.

طبقه‌بندی با استفاده از ویژگی‌های استخراج شده

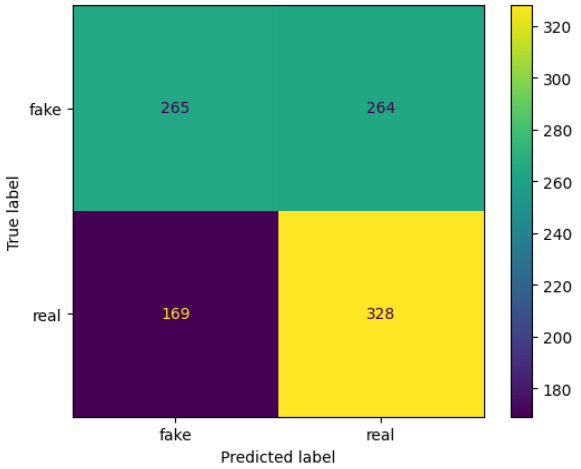
در این بخش تمامی مدل‌های آموزش دیده مرحله قبل به همراه روش AdaBoost مورد آزمایش قرار گرفته‌اند. AdaBoost استفاده شده، از ۱۰۰۰ مدل درخت تصمیم بدون در نظر گرفتن حداکثر عمق بهره می‌برد. همچنین مدل تعداد و اندازه‌ی لایه‌های شبکه عصبی MLP مطابق آنچه در بخش قبل توضیح داده شد، برابر با ۳ لایه با ۴۰۰، ۲۰۰ و ۱۰۰ در نظر گرفته شده است. نتایج مدل‌های بررسی شده به شرح زیر است:

مدل	مجموعه داده	Precision	Recall	F1-Score	Accuracy
Naïve Bayes	آموزش	۰.۶۱۵۶	۰.۶۱۲۱	۰.۶۰۹۷	۰.۶۱۳۱
	تست	۰.۵۸۲۳	۰.۵۸۰۵	۰.۵۷۶۴	۰.۵۷۸۰
Logistic Regression	آموزش	۰.۸۰۵۷	۰.۸۰۵۳	۰.۸۰۵۴	۰.۸۰۵۵
	تست	۰.۵۶۳۸	۰.۵۶۳۸	۰.۵۸۳۴	۰.۵۸۳۴
SVM	آموزش	۰.۹۱۵۴	۰.۹۱۱۲	۰.۹۱۱۵	۰.۹۱۱۸
	تست	۰.۶۲۵۰	۰.۶۲۴۸	۰.۶۲۳۷	۰.۶۲۳۸
MLP	آموزش	۱	۱	۱	۱
	تست	۰.۶۲۲۶	۰.۶۲۲۶	۰.۶۲۱۸	۰.۶۲۱۸
KNN	آموزش	۰.۷۰۲۱	۰.۶۸۸۵	۰.۶۸۴۳	۰.۶۹۰۱
	تست	۰.۵۵۸۹	۰.۵۵۴۲	۰.۵۴۲۴	۰.۵۴۹۷
Decision Tree	آموزش	۱	۱	۱	۱
	تست	۰.۵۵۰۰	۰.۵۴۹۸	۰.۵۴۸۶	۰.۵۴۸۷
AdaBoost	آموزش	۱	۱	۱	۱
	تست	۰.۵۶۸۳	۰.۵۶۸۲	۰.۵۶۷۲	۰.۵۶۷۳

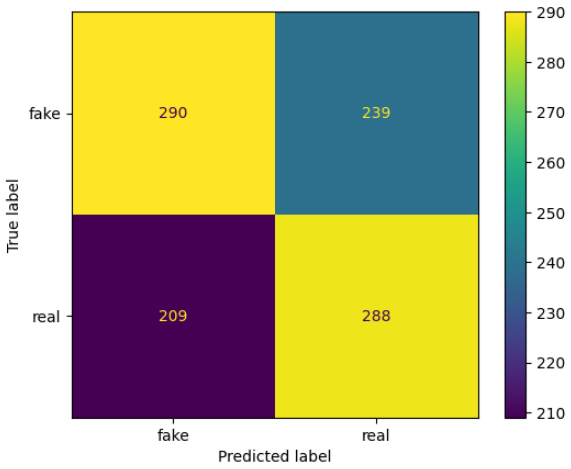
مطابق جدول فوق به نظر می‌رسد تمامی مدل‌های استفاده شده، دچار بیش‌برازش شده‌اند. در برخی مدل‌ها مانند MLP، درخت تصمیم و AdaBoost این مشکل بیشتر است و در بقیه مدل‌ها کمتر به چشم می‌آید. به نظر می‌رسد کم بودن دقت مدل‌های ساده‌ای مانند Naïve Bayes، Logistic Regression، KNN و درخت تصمیم بر روی دادگان موجود، ناشی از خطای بایاس باشد، چرا که با استفاده از مدل‌های پیچیده‌تر مانند MLP و SVM شاهد افزایش دقت و سایر معیارهای ارزیابی بر روی دادگان تست هستیم. اما با پیچیده‌تر شدن مدل، خطای واریانس دامن‌گیر مدل‌ها شده و برای جلوگیری از تاثیرگذاری خطای واریانس بر روی مدل‌ها، نیاز به داده‌های بیشتری برای آموزش وجود دارد که با توجه به ماهیت وظیفه[۱۲] تعریف شده و دشواری دسترسی به داده‌های مشابه و گسترش مجموعه داده، متاسفانه شرایط استفاده از داده‌های بیشتر وجود ندارد.

در ادامه ماتریس آشفتگی هر یک از مدل‌ها بر روی داده‌های تست به نمایش درآمده است:

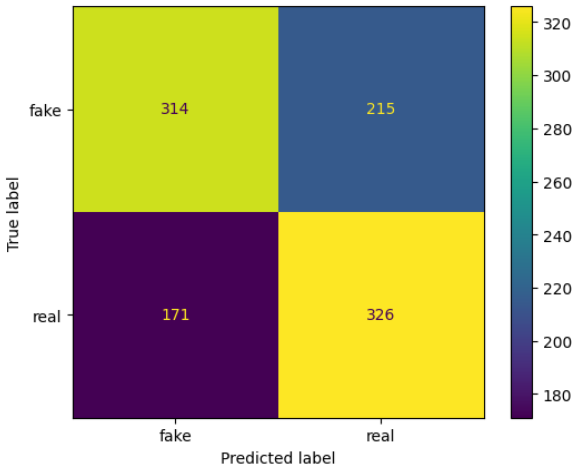
[12] Task



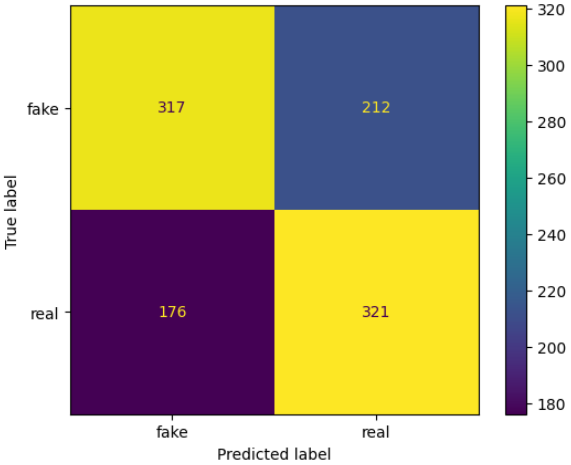
Naïve Bayes



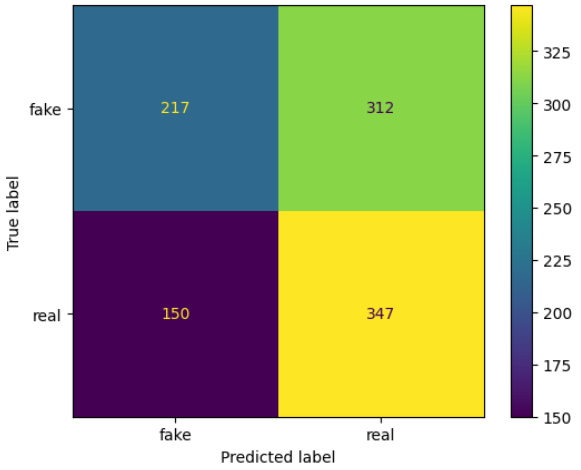
Logistic Regression



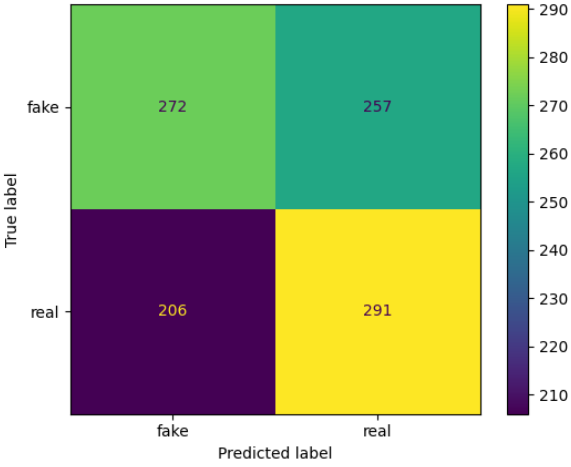
SVM



MLP



KNN



Decision Tree

با توجه به تصاویر فوق، تمامی مدل‌ها بر روی کلاس Real عملکرد بهترین از خود نشان داده‌اند که این عملکرد بهتر به خصوص در مدل‌های KNN و Naïve Bayes مشخص است. همچنین برخی مدل‌ها مانند Naïve Bayes داده‌های کلاس Fake را به کلی تشخیص نمی‌دهند و با دقتی برابر با یک مدل تصادفی این داده‌ها را برچسب می‌زنند.

خوشه بندی

خوشه‌بندی یکی از روش‌های بدون نظارت[۱۳] است که با توجه به فاصله‌هایی بین نمونه‌ها تعریف می‌شود، نمونه‌های نزدیک به یکدیگر را در یک خوشه قرار می‌دهد برای این بخش از سه الگوریتم خوشه‌بندی استفاده شده، این الگوریتم‌ها عبارت‌اند از Gaussian mixture model، KMeans و الگوریتم مبتنی بر چگالی یا DB Scan که نتایج و تحلیل هر الگوریتم در ادامه آورده شده است.

همچنین برای صحت سنجی از کیفیت عملکرد خوشه‌ها از سه معیار ارزیابی استفاده شده است. این معیارها عبارتند از:

۱. خلوص هر خوشه (purity)

این معیار بررسی می‌کند که هر خوشه چه درصدی از نمونه‌های اکثریت مربوط به هر کلاس را به خود اختصاص داده، هرچه این خالص درجه‌ی خلوص هر خوشه بیشتر باشد میتوان نتیجه گرفت که عملکرد مدل بهتر بوده است.

۲. معیار ارزیابی سیلهوت (Silhouette Score)

این معیار با در نظر گرفتن فاصله‌ی نمونه‌ها در هر خوشه و فاصله‌ی خوشه‌ها در هر مدل کیفیت خوشه‌بندی را ارزیابی می‌کند هر چه ای معیار مقدار کمتری داشته باشد نشان‌دهنده کیفیت بالای خوشه‌ها است

۳. فاصله‌ی درون خوشه‌ای

یکی از معیارهایی که میتواند نشان‌دهنده خوشه‌بندی مناسب باشد این که، خوشه‌ها تا حد امکان متراکم باشند.

تمامی معیارهای گفته شده در حالتی که تعداد خوشه‌ها را به اندازه تعداد نمونه افزایش دهیم بهترین نتیجه را ارائه می‌کند ولی برای این که خوشه‌بندی فرم منطقی داشته باشد باید از روشی استفاده کنیم که بتوانیم حد آستانه‌ی مناسبی را برای تعداد خوشه‌ها پیدا کنیم. کی از روش‌های مورد استفاده روش آرنج است که در ادامه برای هر کدام از مدل‌ها با استفاده از آن بهترین خوشه را مشخص می‌کنیم.

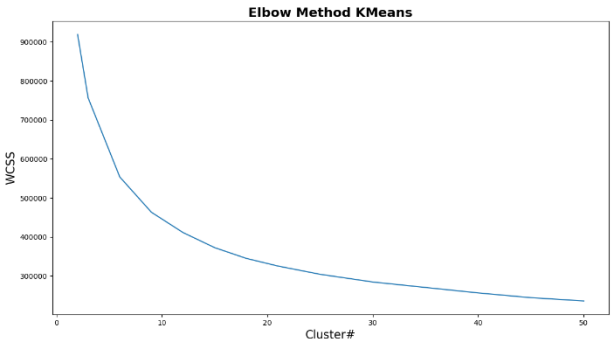
K Means

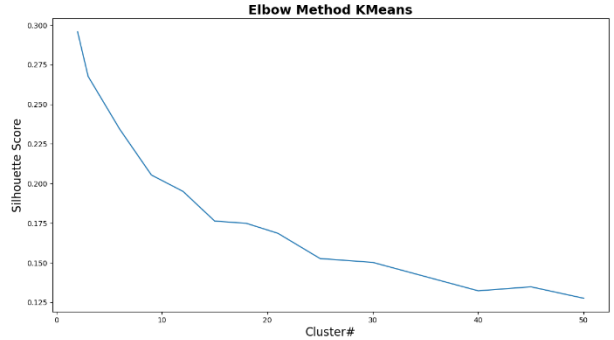
عملکرد این الگوریتم به طور خلاصه به این صورت است که ابتدا تعدادی نقطه‌ی تصادفی (به تعداد خوشه‌های انتخاب شده) در فضای نمونه‌ها انتخاب می‌کند سپس هر نمونه را به نزدیک ترین مرکز اختصاص می‌دهد، نمونه‌ها اختصاص یافته به هر نقطه یک خوشه را تشکیل می‌دهند، این بار مرکز هر خوشه که همان نقطه‌های آغازین بودند بروزرسانی می‌شوند و این روند تا زمانی که نقطه‌ها جا به جا نشوند یا نمونه‌ای از خوشه‌ای به خوشه‌ی دیگر اختصاص نیابد ادامه دارد.

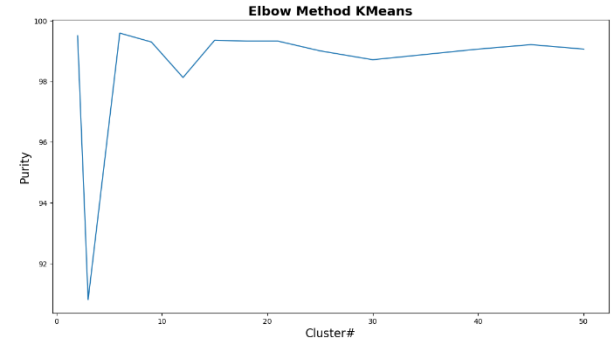
برای بررسی تعداد مناسب خوشه‌ها، دید ابتدایی به این گونه است که به تعداد کلاس‌هایی که نیاز به تشخیص آن‌ها داریم خوشه در نظر بگیریم، اما این فرض همیشه درست نیست به این علت که ممکن است اعضای هر کلاس به جای این که همه در یک خوشه تجمیع شده باشند چند خوشه‌ی غیر متمرکز داشته باشند. به همین علت می‌توانیم تعداد خوشه‌ها را بیشتر از تعداد کلاس‌ها در نظر بگیریم و سپس به جای این که یک خوشه نماینده یک کلاس باشد برای تشخصیص چند خوشه را به یک کلاس تخصیص دهیم.

[13] Unsupervised

با استفاده از ویژگی‌ها آماده عملکرد این مدل به صورت زیر است:







Clusters#	WCSS	Silhouette score	Purity (%)
2	918891.5	0.29576	99.50
3	756252.6	0.267586	90.81
6	553595	0.234216	99.59
9	463108.1	0.20533	99.29
12	411065.3	0.194995	98.12
15	372390.1	0.176268	99.35
18	344814.4	0.174838	99.32
21	325208.9	0.168529	99.32
25	304071.1	0.152593	99
30	284306	0.150158	98.71
40	256274.4	0.13222	99.06
45	244096.1	0.134736	99.21
50	235762.7	0.127525	99.06

در جدول بالا مشخصات هایپرپارامتر مربوط به هر خوشه‌بندی که تعداد خوشه‌ها و در کنار آن معیارهای ارزیابی برای هر کدام از مدل‌ها آورده شده است.

با توجه به متد آرنج و معیارهای Silhouette score و WCSS بهترین تعداد خوشه تعداد ۹ خوشه است ولی معیار purity بهتری تعداد را عدد ۶ مشخص کرده است با توجه به این که خالص بودن خوشه‌ها از اهمیت ویژه‌ای برخوردار است در نتیجه بهتری تعداد خوشه برای این الگوریتم مقدار ۶ است.

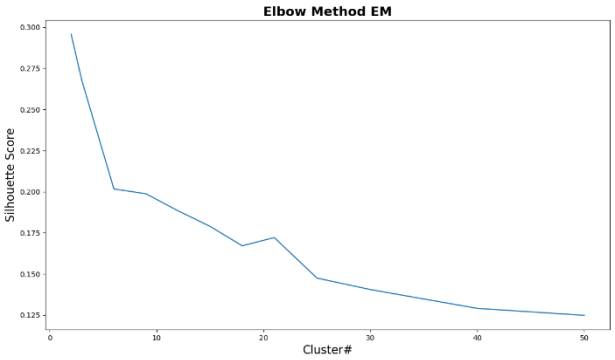
۶ خوشه به این معنی است که مدل توانسته ۶ خوشه خالص را پیدا کند که هر خوشه مختص به یک کلاس خواهد بود. به این ترتیب این امکان وجود دارد که هر کلاس بیش از یک خوشه به عنوان نماینده آن کلاس تخصیص داده شود.

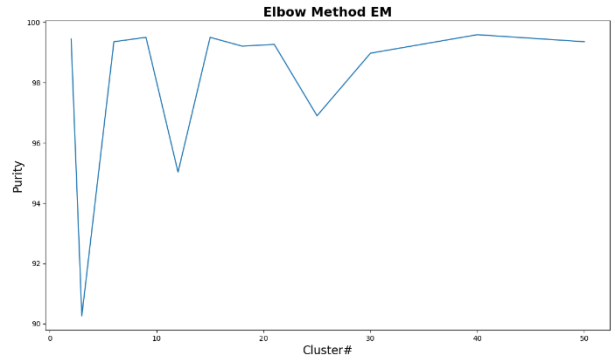
EM GMM

الگوریتم مدل مخلوط گوسی، یک الگوریتم استفاده شده در یادگیری بدون نظارت و خوشه‌بندی است. این الگوریتم به منظور مدل کردن داده‌ها با چندین توزیع گوسی استفاده می‌شود. الگوریتم با شروع از مقادیر تصادفی، پارامترهای مدل مخلوط گوسی را مقداردهی اولیه می‌کند. سپس به صورت تکراری مراحل E و M را انجام می‌دهد. در مرحله E، نقاط داده به توزیع‌های گوسی اختصاص داده می‌شوند براساس احتمالات آنها و در مرحله M، پارامترهای مدل بر اساس اختصاص داده‌ها به‌روزرسانی می‌شوند. الگوریتم تا رسیدن به همگرایی ادامه می‌یابد، یعنی زمانی که پارامترهای مدل ثابت می‌شوند.

نتیجه‌ی آموزش مدل بر روی فیچرهای آماده به صورت زیر است:

Clusters#	Silhouette score	purity
2	0.295718	99.44412
3	0.267592	90.25746
6	0.201606	99.35635
9	0.198666	99.50263
12	0.188328	95.02633
15	0.178835	99.50263
18	0.167042	99.21006
21	0.172049	99.26858
25	0.147396	96.89877
30	0.14042	98.97601
40	0.128982	99.5904
50	0.12475	99.35635



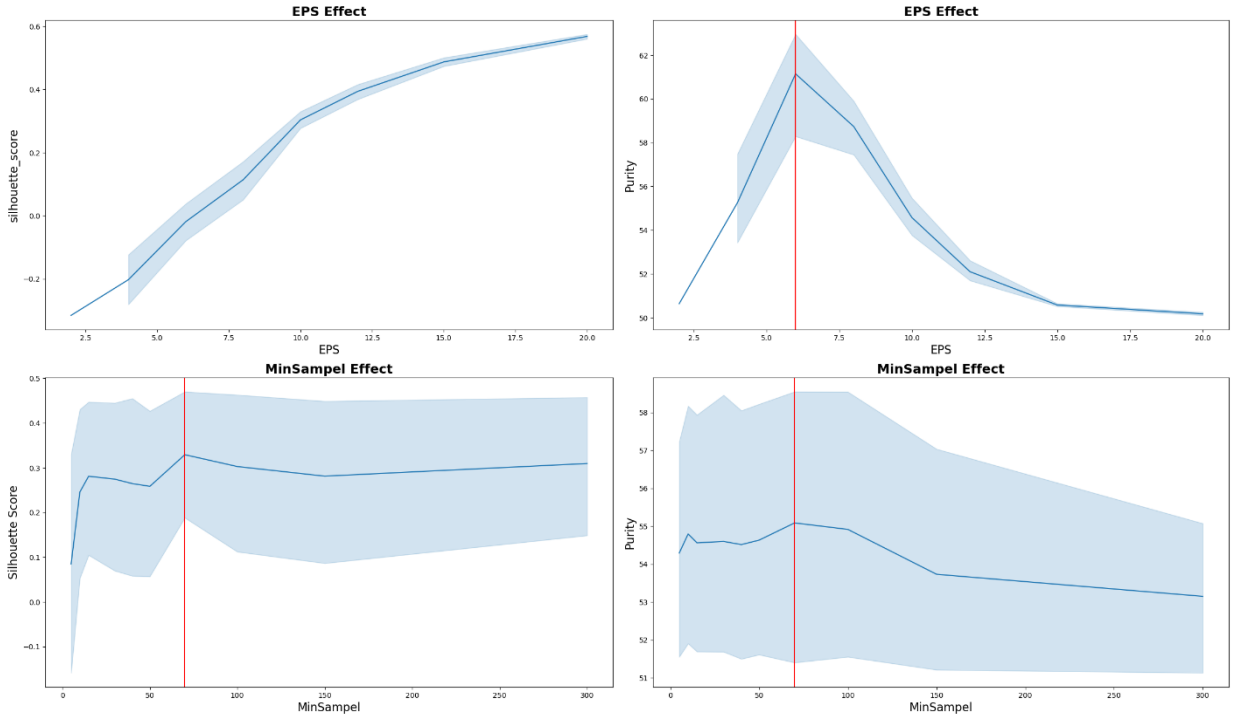


در این گام بهترین تعداد خوشه مقدار ۲ است دلیل آن تنها بر اساس معیار خالص بودن خوشه‌ها است، ترجیح ما همیشه بر این امر است که تعداد خوشه‌ها تا حد امکان کم باشد تا تصمیم گیری برای نمونه‌ی جدید به بهترین نحو انجام شود، حال اگر بخواهیم تصمیمی بر اساس هر دو معیار ارزیابی ارائه کنیم، بهترین تصمیم باز هم عدد ۶ می‌شود چرا که بر روی آرنج نمودار معیار Silhouette هستیم و از طرفی معیار Purity هم مقدار قابل قبولی را ارائه می‌کند.

DB Scan

این الگوریتم یک الگوریتم مبتنی بر چگالی است که می‌تواند خوشه‌ها با فرم‌های غیر دایره‌ای (KMeans) و خوشه‌های غیر بیضوی (GMM) را تشخیص دهد. به طور کلی این الگوریتم می‌تواند خوشه‌هایی به فرم دلخواه را تشخیص دهد. از طرفی این الگوریتم بر خلاف KMeans و GMM نسبت به داده‌های پرت بسیار مقاوم تر است. با تعیین یک شعاع همسایگی (eps) و حداقل تعداد نقاط (min point) مورد نیاز برای تشکیل یک منطقه چگال، کار خود را آغاز می‌کند. این الگوریتم نقاط هسته‌ای را با داشتن تعداد کافی از همسایگان تشخیص می‌دهد و به صورت بازگشتی خوشه‌ها را با اتصال نقاطی که قابل دسترسی هستند گسترش می‌دهد. نقاط مرزی به خوشه‌های همسایه اختصاص داده می‌شوند و نقاط نویزی به عنوان پرتی‌ها شناخته می‌شوند.

نتیجه‌ی این الگوریتم بر رو فیچرهای آماده به صورت زیر است:



ابتدا لازم به ذکر است که این الگوریتم نتوانسته به طور مناسبی خوشه‌بندی را انجام دهد، ولی برای یافتن مقدار بهینه‌ی برای این الگوریتم ابتدا به نموداری که بر اساس معیار Silhouette است دقت می‌کنیم تعداد حداقل نمونه‌ها بعد از مقدار ۷۰ تاثیری در این امتیاز ندارد و عرض ایجاد شده در نمودار نتیجه‌ی تغییرات شعاع همسایگی است. در نتیجه بیشترین مقداری که می‌تواند نتیجه‌ی این معیار را بهبود دهدمقدار ۷۰ است که در نمودار purity نیز این عدد نتیجه می‌شود چرا که در مقدار ۷۰ این معیار مقدار خوبی دارد. برای تعیین شعاع همسایگی با توجه به معیار purity مقدار ۶ که با خط قرمز مشخص شده است بهترین مقدار است و از روی نمودار Silhouette بهترین مقدار، عدد ۶ است، البته مقادیر ۸ و ۱۰ نیز می‌توانند مناسب باشند، ولی با توجه به این معیار در مقدار ۶ کمتر شده است همین عدد را به عنوان عدد بهینه انتخاب می‌کنیم تعداد خوشه‌های تشکیل شده با شعاع ۶ و حداقل نمونه ۷۰ عدد ۲ است، یعنی دو خوشه تشکیل شده است. (فایل خروجی این بخش در کنار سایر فایل‌ها قرار داده شده است *DBScan_res.csv*)

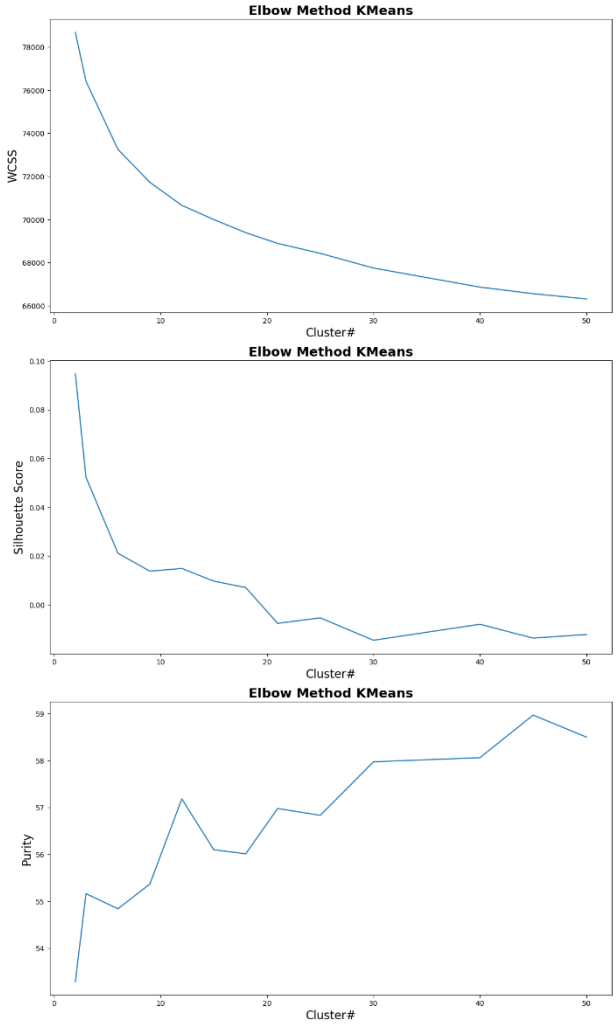
خوشه بندی بر روی ویژگی‌های استخراج شده

ویژگی HOG

پس استخراج ویژگی انتخاب شده که توضیحات آن در بخش قبل ارائه شد الگوریتم‌ها خوشه بندی را به ترتیب بر روی آن اجرا کردیم

الگوریتم KMeans

نتایج این الگوریتم به صورت زیر است:

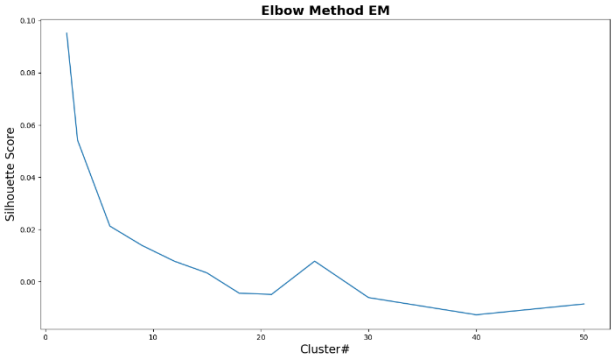


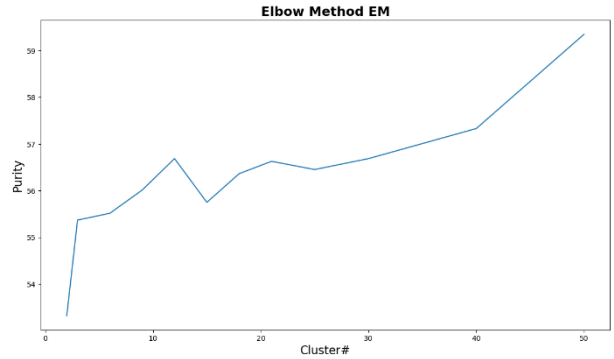
Clusters#	WCSS	Silhouette score	Purity(%)
2	78674.02	0.094738	53.29
3	76408.43	0.052316	55.16
6	73251.01	0.021191	54.84
9	71724.37	0.01384	55.37
12	70657.97	0.014958	57.18
15	69999.42	0.009764	56.10
18	69391.24	0.00714	56.01
21	68893.53	-0.00754	56.97
25	68430.67	-0.0053	56.83
30	67750.4	-0.01447	57.97
40	66865.75	-0.00791	58.06
45	66560.54	-0.0136	58.96
50	66318.15	-0.01211	58.50

بعد از خوشه بندی با استفاده از الگوریتم KMeans حد آستانه‌ای که برای خوشه‌بندی این ویژگی‌انتخاب شده مقدار ۱۲ خوشه است. که در این تعداد خوشه نمودار Silhouette و Purity اندکی برآمدگی دارد نسبت به نقطه‌ی قبل خود.

الگوریتم EM GMM

در این بخش با استفاده از الگوریتم خوشه‌بندی توزیع مخلوط گوسی خوشه بندی انجام شد که نتایج آن به صورت زیر است:



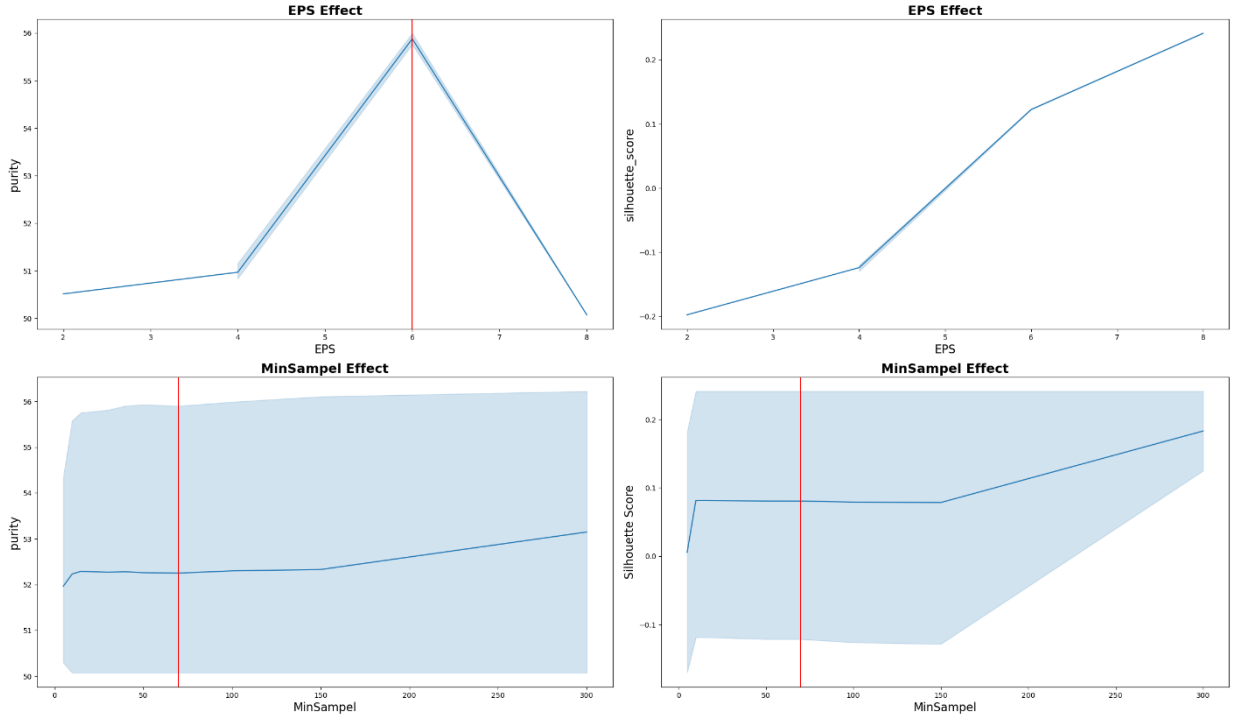


Clusters#	Silhouette score	purity
2	0.0951	53.32
3	0.054188	55.37
6	0.021256	55.51
9	0.01381	56.01
12	0.007785	56.68
15	0.003354	55.75
18	-0.004432	56.36
21	-0.004897	56.62
25	0.007831	56.45
30	-0.006116	56.68
40	-0.012702	57.33
50	-0.008555	59.35

تعداد کامپوننت مناسب برای این بخش توسط ۱۲ کامپوننت بدست آمده است. به طور کلی بهتر است که معیار purity برای ارزیابی خوشه‌ها استفاده شود چرا که این ارزیابی دانش برچسپ‌های نمونه‌ها را نیز مداخله می‌دهد.

الگوریتم DBSCAN

نتایج این الگوریتم بر روی ویژگی HOG به صورت زیر است:



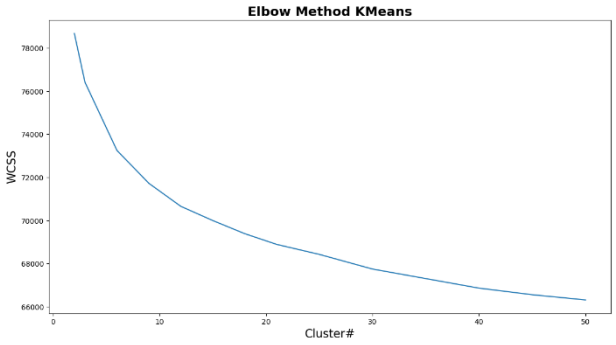
با توجه به نمودار مقدار بهینه برای مقدار شعاع عدد ۶ است که purity در آن نقطه افزایش مناسبی داشته، اما مورد مقدار بهینه برای تعداد حداقل نمونه‌ها وضعیت در بین مقادیر ۳۰ تا ۱۵۰ برای هر دو معیار یکسان است ولی برای این که حد فاصل مناسبی را برای آموزش مدل رعایت کنیم بهتر است که مقداری در بین این اعداد انتخاب شود مقدار انتخاب شده برای این بخش به عنوان هاپرپارامتر بهینه عد ۷۰ است.

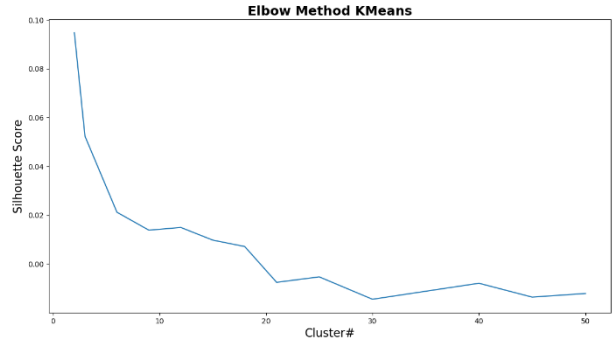
ویژگی **LBP**

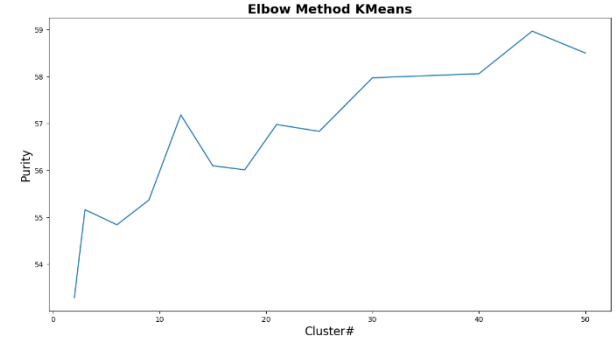
پس استخراج ویژگی انتخاب شده که توضیحات آن در بخش قبل ارائه شد الگوریتم‌ها خوشه بندی را به ترتیب بر روی آن اجرا کردیم

الگوریتم KMeans

نتایج این الگوریتم به صورت زیر است:





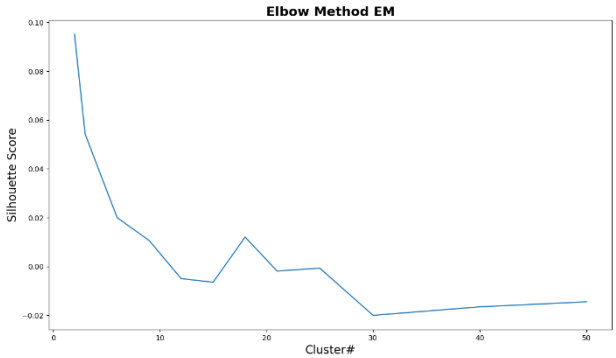


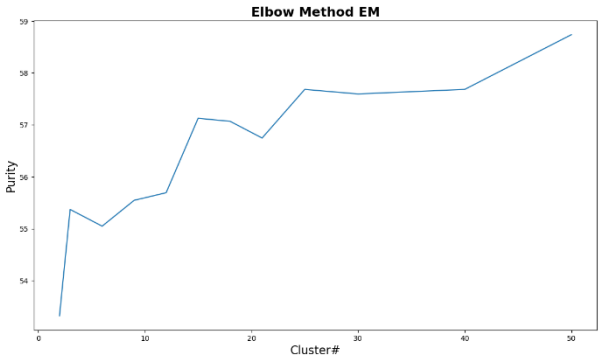
Clusters#	WCSS	silhouette_score	purity
2	999648600	0.019721	50.68
3	993335000	-0.014559	52.12
6	987435400	-0.02073	54.28
9	985653300	-0.025712	54.78
12	982174800	-0.033667	55.13
15	981657400	-0.030841	53.64
18	976158200	-0.030158	54.17
21	976663400	-0.029159	54.11
25	974365000	-0.039322	54.98
30	970007300	-0.035544	54.14
40	966345100	-0.032175	55.63
45	963181700	-0.035629	54.9
50	960152600	-0.032464	56.65

در این قسمت اگر تعداد خوشه‌ها بر اساس purity و حد آستانه‌ای برای این که تعداد خوشه‌هایمان زیاد نظر بگیریم بهترین تعداد خوشه‌ها عدد ۱۲ خواهد بود ولی اگر بخوایهم معیار silhouette را نیز در نظر بگیریم، مقدار ۹ نیز می‌تواند مناسب باشد.

الگوریتم EM GMM

در این بخش با استفاده از الگوریتم خوشه‌بندی توزیع مخلوط گوسی خوشه بندی انجام شد که نتایج آن به صورت زیر است:



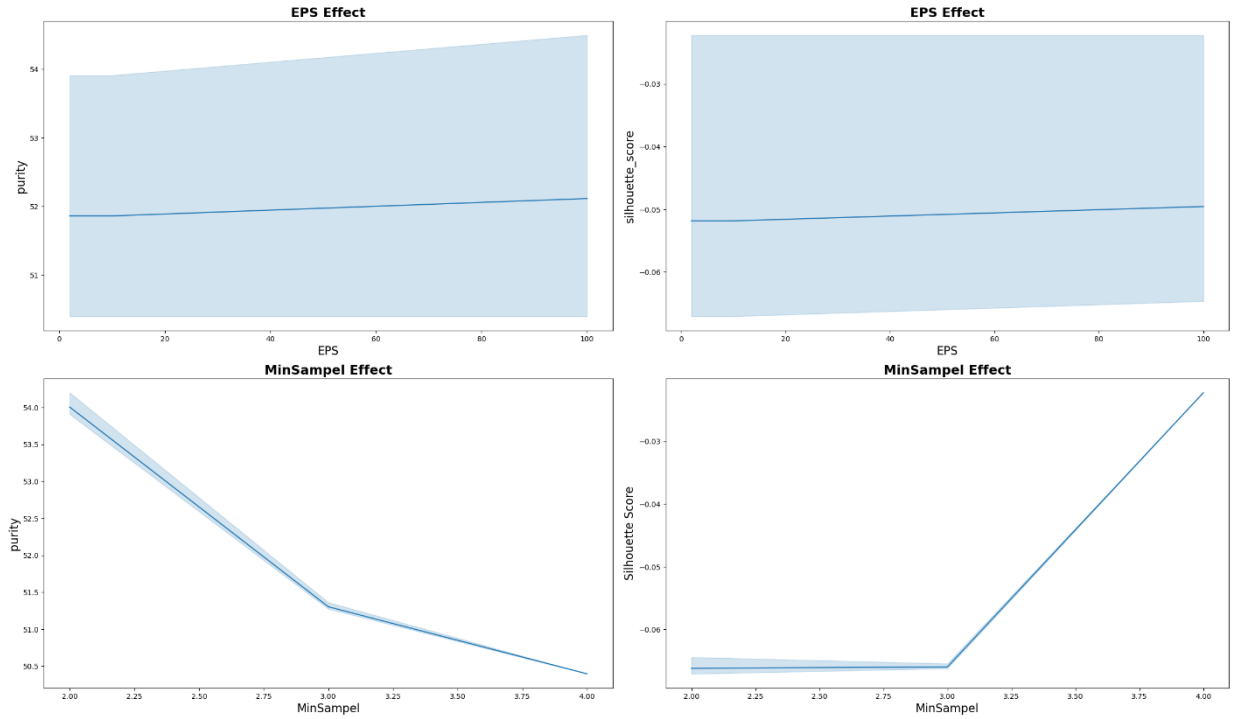


Clusters#	Silhouette score	purity
2	0.0951	53.32
3	0.054188	55.37
6	0.021256	55.51
9	0.01381	56.01
12	0.007785	56.68
15	0.003354	55.75
18	-0.00443	56.36
21	-0.0049	56.62
25	0.007831	56.45
30	-0.00612	56.68
40	-0.0127	57.33
50	-0.00856	59.35

تعداد کامپوننت مناسب برای این بخش توسط ۱۵ کامپوننت بدست آمده است.به این دلیل که روند افزایش درجه‌ی خلوص و کاهش معیار Silhouette تا این مقدار شیب مناسبی دارد و پس از آن بهتر است که از بهبود بیشتر این معیارها به دلیل overfit بپرهیزیم.

الگوریتم DBSCAN

نتایج این الگوریتم بر روی ویژگی LBP به صورت زیر است:



در این بخش به دلیل متعدد بودن ابعاد این ویژگی کارایی الگوریتم مبتنی بر چگالی به شدت ضعیف شده به طوری که به سختی می‌تواند تعدادی کلاستر تشکیل دهد و با کاهش تعداد نقاط حداقلی برای نقاط هسته‌ای و افزایش شعاع تمامی نمونه‌ها را به یک خوشه اختصاص می‌دهد در نتیجه نمی‌توان مقدار مناسبی برای هاپرپارامترهای این مدل انتخاب کرد.