به نام خدا

مستند پروژه پایانی درس مبانی بینایی کامپیوتر

مدرس: دکتر محمدرضا محمدی

دانشگاه علم‌وصنعت ایران

گردآورندگان:

محمدامین رضاپور – ۹۹۵۲۱۳۰۷

مهدی قضاوی – ۹۹۵۲۲۰۱۴

الگورریتم‌های Anti-spoofing:

تشخیص زنده بودن چهره (Liveness Detection) یک روش در حوزه امنیت بیومتریک است که هدف آن تشخیص این است که آیا یک تصویر یا ویدیو از یک چهره واقعی و زنده گرفته شده است یا از یک تصویر، ویدیو، یا مدل سه‌بعدی جعلی استفاده می‌کند. این تکنیک‌ها به منظور مقابله با حملات اسپوفینگ (spoofing) طراحی شده‌اند که در آن مهاجمان سعی می‌کنند با استفاده از عکس، ویدیو یا ماسک‌های چهره‌ای، سیستم‌های شناسایی چهره را فریب دهند.

روش‌های تشخیص زنده بودن چهره

- روش‌های مبتنی بر حرکت (Motion-Based): تشخیص حرکات طبیعی سر و صورت: بررسی حرکت‌های کوچک و طبیعی که به سختی توسط تصاویر یا ویدیوهای جعلی قابل تقلید هستند.
- تکنیک‌های چالش و پاسخ (Challenge-Response): درخواست از کاربر برای انجام یک عمل خاص مانند پلک زدن، لبخند زدن یا چرخاندن سر.
- روش‌های مبتنی بر تحلیل تصویر (Image-Based): تحلیل نور و بازتاب: بررسی الگوهای بازتاب نور در صورت، که در تصاویر دوبعدی و جعلی با تصاویر واقعی متفاوت است.
- بررسی ویژگی‌های ریزپوستی (Micro-Texture Analysis): استفاده از ویژگی‌های پوستی مثل منافذ و بافت‌های ریز که در تصاویر جعلی معمولاً وجود ندارد.
- روش‌های مبتنی بر داده‌های چندبعدی (Depth-Based): استفاده از دوربین‌های سه‌بعدی: این دوربین‌ها می‌توانند عمق و ساختار سه‌بعدی صورت را تشخیص دهند، که در تصاویر دوبعدی قابل تقلید نیست.
- استفاده از تکنیک‌های بازسازی سه‌بعدی: استفاده از اطلاعات چند زاویه‌ای برای بازسازی و تحلیل ساختار سه‌بعدی چهره.
- روش‌های مبتنی بر یادگیری ماشین (Machine Learning-Based): شبکه‌های عصبی و یادگیری عمیق: استفاده از شبکه‌های عصبی عمیق برای آموزش مدل‌های تشخیص زنده بودن که می‌توانند الگوهای پیچیده‌ای را که نشان‌دهنده زنده بودن هستند، شناسایی کنند.

- مدل‌های ترکیبی: ترکیب چندین روش فوق برای ایجاد یک سیستم قوی‌تر و دقیق‌تر.

کاربردهای تشخیص زنده بودن چهره

- سیستم‌های احراز هویت بیومتریک: برای جلوگیری از ورود غیرمجاز به سیستم‌های کامپیوتری، دستگاه‌های تلفن همراه، یا حساب‌های آنلاین.
- کنترل دسترسی: در مکان‌های حساس مانند بانک‌ها، فرودگاه‌ها و سازمان‌های دولتی.
- پرداخت‌های موبایلی و بانکی: برای افزایش امنیت تراکنش‌های مالی آنلاین و موبایلی.
- سیستم‌های نظارت و امنیت: برای تشخیص حضور واقعی افراد در سیستم‌های نظارتی و امنیتی.

تشخیص زنده بودن چهره یک حوزه پویا و در حال توسعه است که با پیشرفت‌های تکنولوژی و ظهور روش‌های جدید حملات، نیازمند بهبود و ارتقاء مستمر است.

هدف پروژه:

در این پروژه، قصد داریم یک الگوریتم Anti-spoofing را برای تشخیص زنده‌بودن (Liveness) چهره در یک ویدیو ورودی توسعه دهیم که در آن، یک فریم از ویدیو را گرفته و تشخیص دهد تصویر زنده شخص است یا خیر (Live/Spoof). درنهایت، کلاسی که در بین فریم‌های یک ویدیو تعداد بیش‌تری برچسب بخورد به عنوان برچسب کل ویدیو انتخاب شده و امتیاز آن برابر با میانگین امتیاز تمامی فریم‌هایی از این ویدیو که چنین برچسبی دارند خواهد بود.

در این سیستم، دو نوع مدل برای این هدف پیاده کرده و آموزش دادیم. مدل اول، یک مدل مبتنی بر ساختار یادگیری عمیق بوده و مدل دوم نیز ابتدا تعدادی ویژگی مهندسی شده از هرفریم استخراج کرده و سپس این ویژگی‌ها را به‌عنوان ورودی به شبکه می‌دهد.

ارزیابی مدل‌های آموزش دیده:

برای ارزیابی این مدل‌ها، ۱۰ مورد ویدیو تست شامل حالت‌های مختلف spoof (غیرزنده) و non-spoof (زنده) جمع‌آوری کردیم تا عملکرد مدل‌های آموزش‌دیده را برروی این داده‌ها ارزیابی کنیم.

مرحله اول: جمع‌آوری مجموعه داده

درمرحله اول، برای ساختن یک سیستم Anti-spoofing که عملکرد قابل‌قبولی برروی تصاویر ورودی داشته باشد، نیاز به جمع‌آوری یا پیداکردن دیتاستی برای همین تسک داریم تا بتوان ساختار مدلی که پیاده می‌کنیم را برروی آن آموزش داد.

از بین دیتاست‌های پیشنهادی در مستند پروژه، نمونه جامع و کاملی به‌صورت رایگان یافت نشد، بنابراین با کمی جست‌وجو، دیتاست معروف CelebA-spoof را پیدا کردیم که به‌صورت رایگان در دسترس عموم قرار دارد. البته که حجم کلی این این دیتاست بسیار حجیم بوده و درحدود ۷۷ گیگ می‌باشد اما باوجود قرارداشتن این دیتاست درفضای سایت Kaggle، توانستیم آن‌را به‌راحتی در نوت‌بوک مربوط به هرمدل از پروژه لود کرده و از این دیتاست کامل و جامع استفاده کنیم.

البته که استفاده از کل این دیتاست برای آموزش شبکه عصبی پیاده‌شده، عملا به‌علت حجم و تعداد بالای داده‌های آموزشی این دیتاست و محدودیت منابع محاسباتی دردسترس از جمله GPUهای موجود در سایت Kaggle و قطع‌شدن Draft Sessionهای این سایت، امکان پذیر نبوده‌است. بااین حال، برای انجام این پروژه، از سیاست Down Sampling استفاده کرده و یک Fraction از دیتاست کلی را جدا کرده و به‌عنوان دیتای آموزشی تعریف می‌کنیم.

مدل نوع اول: مدل مبتنی بر یادگیری عمیق

برای بررسی این مدل، نوت‌بوک الصاق شده با نام FaceAntiSpoofingWithMobileNetV2.ipynb را شرح می‌دهیم.

آماده‌سازی داده‌ها: همانطور که گفته شد از یک نسخه Down Sampleشده دیتاست CelebA-spoof استفاده کرده و مجموعه‌داده‌های train/val/test را تعریف می‌کنیم. هم‌چنین transformationهای زیر را برروی هر یک از این داده‌ها اعمال می‌کنیم:

```
# transformations
transforms = {
    'train': transforms.Compose([
        transforms.RandomResizedCrop(224),
        transforms.RandomHorizontalFlip(),
        transforms.RandomRotation(10),
        transforms.ToTensor(),
        transforms.Normalize(
            mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]
        )
    ]),
    'test': transforms.Compose([
        transforms.Resize(256),
        transforms.CenterCrop(224),
        transforms.ToTensor(),
        transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225])
    ])
}
```

در ادامه، در کلاس تنظیم‌شده FASDataset، علاوه‌بر هرداده و برچسب آن، برای هرتصویر تبدیل فوریه آن را نیز حساب کرده و درکنار تصویر برمی‌گردانیم چرا که در بخش آموزش شبکه، برای محاسبه تابع ضرر مدل، به آن نیاز خواهیم داشت.

مدل عمیقی که ساختار آن را پیاده‌کردیم، شامل دو بلوک classification و Fourier Prediction می‌باشد. در بلوک classification، backbone شبکه را به‌کمک مدل MobileNetV2 تعریف کرده و وزن‌های pretrain این مدل برروی imagenet را استفاده می‌کنیم، سپس یک لایه AvgPooling، یک لایه Flatten و درنهایت یک لایه Dense برای دسته‌بندی دوکلاسه به انتهای آن اضافه می‌کنیم.

در بلوک تبدیل فوریه، از ساختار استفاده شده در مدل Silent Anti-spoofing استفاده کردیم که با تعدادی بلوک‌های خطی، کانولوشنی، Depthwise کانولوشن و Residual، ابتدا سعی می‌کند تعدادی ویژگی از تصویر ورودی استخراج کرده و سپس به‌کمک بلوک FTGenerator

تبدیل فوریه تصویر را حساب کند. در زیر، علت استفاده از تبدیل فوریه و ویژگی‌های آن در این نوع سیستم‌ها را بررسی کوتاهی می‌کنیم:

تحلیل ویژگی‌های فرکانسی:

- تفاوت‌های فرکانسی: تصاویر واقعی و تصاویر جعلی تفاوت‌های فرکانسی مشخصی دارند. برای مثال، تصاویر واقعی معمولاً دارای جزئیات و بافت‌های پیچیده‌تری هستند که در فرکانس‌های بالا ظاهر می‌شوند، در حالی که تصاویر چاپ شده یا نمایش داده شده روی صفحه نمایش ممکن است این جزئیات را نداشته باشند.
- تشخیص الگوهای تکراری: تصاویر جعلی ممکن است الگوهای تکراری و مصنوعی داشته باشند که در حوزه فرکانس به وضوح قابل تشخیص هستند.
- کاهش نویز و بهبود کیفیت داده:
- فیلتراسیون فرکانسی: تبدیل فوریه می‌تواند برای حذف نویز و بهبود کیفیت تصاویر مورد استفاده قرار گیرد. این به سیستم‌های یادگیری عمیق کمک می‌کند تا با داده‌های تمیزتر و با کیفیت‌تری آموزش ببینند.
- تمرکز بر اطلاعات مهم: با تحلیل داده‌ها در حوزه فرکانس، می‌توان اطلاعات غیرضروری را حذف کرده و فقط به بخش‌های مهم‌تر تمرکز کرد.

استخراج ویژگی‌ها:

- تبدیل داده‌های زمانی-مکانی به داده‌های فرکانسی: بسیاری از ویژگی‌های مهم در تصاویر در حوزه فرکانس بهتر قابل تشخیص هستند. تبدیل فوریه می‌تواند به استخراج ویژگی‌های موثرتر برای مدل‌های یادگیری عمیق کمک کند.
- کاهش ابعاد: با استفاده از تبدیل فوریه، می‌توان داده‌های با ابعاد بالا را به فضای فرکانس تبدیل کرد و سپس فقط اجزای مهم‌تر را نگه داشت، که این کار به کاهش ابعاد داده‌ها و افزایش کارایی مدل‌های یادگیری کمک می‌کند.

افزایش مقاومت در برابر حملات:

- تشخیص الگوهای جعلی: بسیاری از حملات اسپوفینگ الگوهای مشخص و قابل شناسایی در حوزه فرکانس دارند. با استفاده از تبدیل فوریه، می‌توان این الگوها را بهتر شناسایی و از بین برد.
- تشخیص تغییرات ناپیوسته: حملات جعلی معمولاً تغییرات ناپیوسته و غیرطبیعی در تصاویر ایجاد می‌کنند که در حوزه فرکانس بهتر قابل تشخیص هستند.

بهینه‌سازی مدل یادگیری عمیق: برای آموزش مدل، تابع ضرر را به‌صورت ترکیبی از خروجی مدل برای classification و Fourier Transform تعریف می‌کنیم به‌همین علت از تابع ضررهای CrossEntropyLoss و MSELoss استفاده می‌کنیم. هم‌چنین از بهینه‌ساز AdamW برای آپدیت‌کردن پارامترهای مدل در هر مرحله استفاده کرده‌ایم.

```
        loss_cls = criterion_cls(outputs_cls, labels)
        loss_ft = criterion_ft(outputs_ft, ft_inputs)
        loss = 0.5 * loss_cls + 0.5 * loss_ft
        loss.backward()
        optimizer.step()
        running_loss += loss.item()

        preds = torch.argmax(outputs_cls, dim=1)
        running_acc += (preds == labels).sum().item()
```

مدل را به‌تعداد ۸۰ ایپوک برروی کسری از داده‌های دیتاست CelebA-spoof آموزش دادیم که درنهایت دقت آن برروی یک Down Sample رندوم از این دیتاست به‌صورت زیر بدست آمد:

```
# Evaluate the model
test_model(model, dataloader_test, device)
```

```
100%|██████████| 57/57 [01:10<00:00,  1.24s/it, acc=0.67]
Accuracy: 67.00%
Precision: 0.6326, Recall: 0.6700, F1 Score: 0.5422
```

درانتهای این نوت‌بوک نیز سه تابع جهت ارزیابی مدل و مشاهده Prediction آن درحالت‌های تصویر Cropشده چهره، کل تصویر و تبدیل فوریه تصویر پیاده‌شده‌اند.

مدل نوع دوم: مدل مبتنی بر استخراج ویژگی‌های مهندسی‌شده

ساختار و آموزش این مدل در فایل نوت‌بوک face-anti-spoofing-with-mobilenetv-2-feature-base.ipynb انجام‌شده است. این مدل را نیز برروی دیتاست قبلی آموزش دادیم. هم‌چنین همانند مدل قبل، به‌کمک تابع extrat_face() و البته cv2.CascadeClassifier() چهره درون تصویر را استخراج کرده و مدل را برروی آن آموزش می‌دهیم.

در ساختار این مدل، ابتدا از تصویر ورودی، سه ویژگی LBP، تبدیل فوریه تصویر و تصویر در فضای HSV استخراج شده و سپس این ویژگی‌ها را به‌عنوان ورودی، به ساختار مدل عمیق مرحله قبلی می‌دهیم.

سایر موارد از قبیل پارامترهای بهینه‌سازی و آموزش و ساختار مدل همانند روش قبلی بوده و درپایان مدل را به‌اندازه ۱۹ ایپوک آموزش داده‌ایم.

ارزیابی مدل و مشاهده خروجی‌ها برروی مجموعه داده جمع‌آوری شده:

درنوت‌بوک main.ipynb، ابتدا مدل‌های تعریف شده بالا را پیاده کرده و سپس وزن‌های آموزش دیده این مدل‌ها را برروی هرکدام load می‌کنیم. درمرحله بعد، چون داده‌های تست از نوع ویدیو هستند، همانطور که درابتدای این مستند اشاره شد، هرفریم را جدا به مدل مربوطه داده تا خروجی آن را بدست آورده و سپس یک خروجی کلی براساس کلیه فریم‌ها برای ویدیو بدست می‌آید. برای ورودی دادن هرفریم نیز از کتابخانه cv2 و متد cv2.VideoCapture() استفاده کرده‌ایم. درپایان

خروجی‌های هر مدل پس از سه روش ارزیابی در فایل های predictions_deep.csv و predictions_feature.csv ذخیره می‌شوند.

```
deep_pred = pd.read_csv('/content/predictions_deep.csv')
deep_pred
```

	filename	liveness_score	liveness_score_crop	liveness_score_frequency
0	spoof1.mp4	0.639763	0.729876	0.767381
1	spoof2.mp4	0.760516	0.734344	0.758207
2	spoof3.mp4	0.664728	0.782748	0.777733
3	spoof4.mp4	0.798873	0.857166	0.767925
4	spoof5.mp4	0.677845	0.748057	0.785354
5	spoof6.mp4	0.730612	0.736595	0.796380
6	non-spoof1.mp4	0.603222	0.665796	0.829088
7	non-spoof2.mp4	0.707813	0.748223	0.787779
8	non-spoof3.mp4	0.718835	0.681930	0.770095
9	non-spoof4.mp4	0.704911	0.662527	0.784996

لینک مربوط به دیتاست تست

منابع استفاده شده:

- دیتاست CelebA-spoof: https://www.kaggle.com/datasets/attentionlayer241/celeba-spoof-for-face-antispoofing/data
- مدل Silent Anti-Spoofing: https://github.com/minivision-ai/Silent-Face-Anti-Spoofing

- وب‌سایت: https://antispoofing.org/face-recognition-methods-complete-overview/
- ویدیو https://www.youtube.com/watch?v=aGSR_3IEIwc

پایان