

گزارش پروژه Appliance Detection

آریا ابراهیمی، سارا قوام‌پور، ملیکا ذبیحی نیشابوری

۱۳ تیر ۱۴۰۲

۱ مقدمه

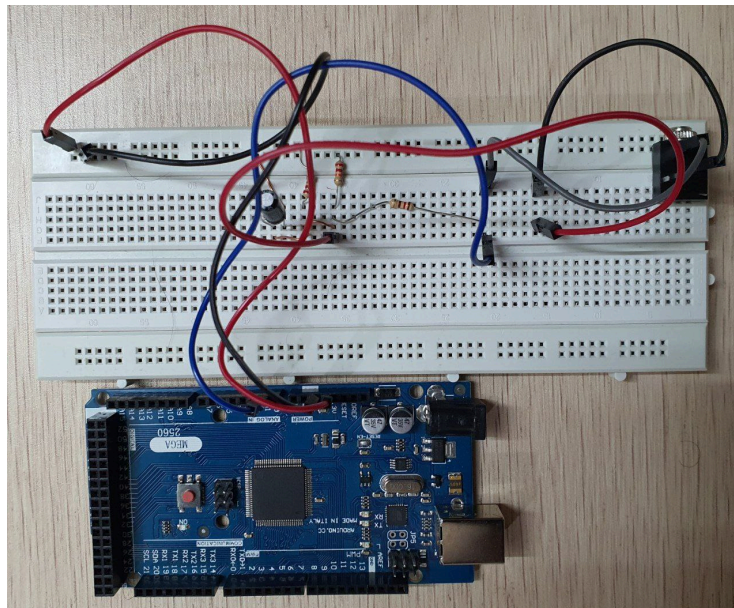
هدف کلی از این پروژه تشخیص رویدادها در شبکه مصرفی خانگی می‌باشد. روش ارائه شده در این پروژه مبتنی بر شبکه‌های عصبی پیچشی^۱ می‌باشد. در بخش اول به منظور جمع‌آوری داده برای آموزش شبکه، نیاز است تا از جریان مصرفی وسایل نمونه برداری شود. در گام بعدی پیش‌پردازش‌های لازم بر روی داده‌های به دست آمده انجام می‌شود. سپس شبکه با داده‌های پیش‌پردازش شده آموزش می‌بیند و در نهایت با وزن‌های به دست آمده توسط شبکه ذکر شده، برای داده‌ها به صورت برخط، رویدادهای مربوط به وسایل را تشخیص می‌دهد.

۲ جمع‌آوری داده

برای این بخش یک سنسور غیر تهاجمی^۲ SCT013 و برد Arduino مورد استفاده قرار گرفته است. سنسور SCT013 باید دور یکی از سیم‌های فاز یا نول قرار بگیرد به صورتی که جهت فلش آن به سمت مصرف کننده باشد. داده‌های مربوط به این سنسور توسط مدار شکل ۱ به سیگنال‌های دیجیتال تبدیل شده و در نهایت به عنوان ورودی به Arduino داده می‌شود. در این پروژه از داده‌های مربوط به سشوار و اتوی مو استفاده شده است، همچنین برای عمل کرد بهتر و قدرت تمایز بیشتر، داده مربوط به عدم رخداد رویداد نیز جمع‌آوری شده است. بدین منظور وسیله‌های انتخاب شده را به یک سهرای که فاز و نول آن جدا شده است متصل کرده و با خاموش و روشن کردن سهرای، جریان مصرفی هر کدام را به دست می‌آوریم. در کد مربوط به Arduino از کتابخانه EmonLib که مربوط به داده‌های انرژی است استفاده شده است. این کتابخانه در هر ثانیه تقریباً ۵۵۸۸ نمونه می‌گیرد. در این پروژه از ۱۱۲ نمونه برای محاسبه جریان RMS استفاده می‌شود. بنابراین در هر ثانیه ۵۰ نمونه متفاوت از جریان RMS به دست خواهد آمد.

CNN^۱

Non-intrusive^۲



شکل ۱: مدار استفاده شده برای دریافت داده‌ها و ارسال به Arduino

۳ پیش پردازش

در این بخش ابتدا مشتق جریان RMS محاسبه می‌شود. در یادگیری عمیق، برای کار با داده‌های به فرم سیگنال، مرسوم است که تبدیل فوریه کوتاه مدت سیگنال برای به دست آوردن ویژگی‌های سیگنال محاسبه شود. Spectrogram حاصل برای رویداد های خاموش و روشن یکسان است، برای رفع این مشکل اسپکتوگرام های به دست آمده در تابع علامت ضرب می‌شوند. تابع علامت با محاسبه تفاضل دو جریان RMS متوالی، صعودی یا نزولی بودن را نشان می‌دهد. مثبت بودن تابع علامت به معنای رویداد روشن شدن است.

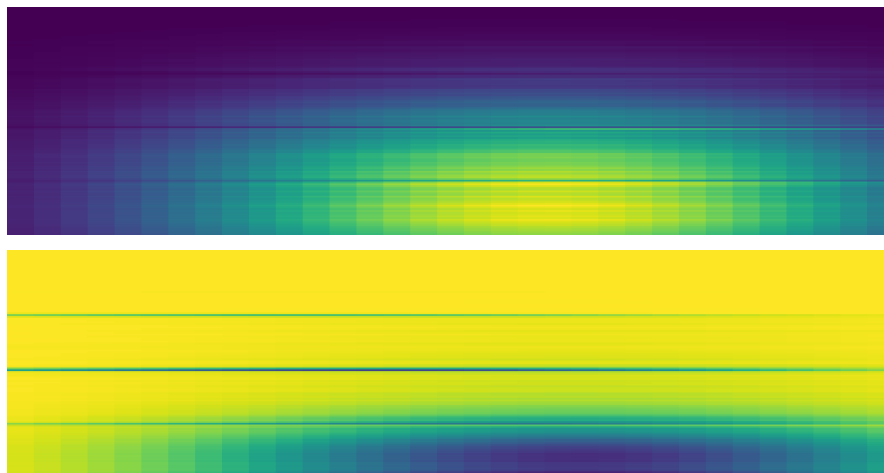
برای جلوگیری از overfit شدن شبکه، از دو مرحله data augmentation به منظور افزایش تعداد داده‌ها استفاده شده است.

در مرحله اول در زمان دریافت سیگنال‌های ورودی، مقدار کمی نویز به آن‌ها اضافه می‌شود. بدین منظور چهار توزیع متفاوت نویز به هر ۵۱۲ نمونه جریان RMS اضافه می‌شوند. عکس‌های به دست آمده با اندازه 32×148 ذخیره می‌شوند.

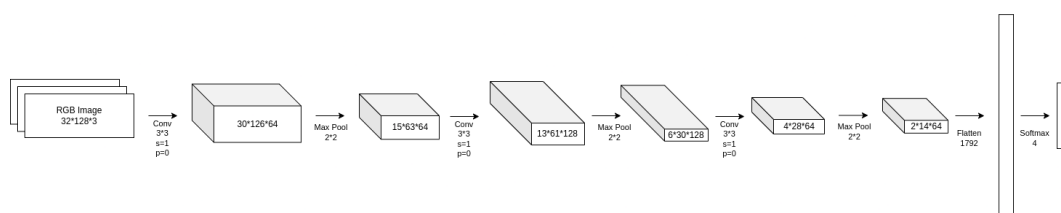
در مرحله دوم، چهار برش تصادفی با اندازه 32×128 از تصاویر قبلی جدا می‌شوند. بدین ترتیب از ۵۱۲ نمونه واقعی، ۲۰ تصویر تشکیل می‌شود.

۴ شبکه عصبی

ورودی شبکه با سایز 32×128 از لایه های مشخص شده در شکل ۳ عبور می‌کند. همچنین از یک لایه Dropout برای جلوگیری از overfitting استفاده شده است.



شکل ۲: در تصویر بالا، اسپکتوگرام به دست آمده مربوط به رویداد روشن شدن سشوار، و در تصویر پایین اسپکتوگرام مربوط به رویداد خاموش شدن سشوار نمایش داده شده اند.



شکل ۳: معماری ارائه شده برای شبکه

تعداد کل داده های استفاده شده برای آموزش شبکه ۵۳۸۰ است، که از این مقدار ۴۳۰۴ برای داده های آموزش و ۱۰۷۶ برای داده های اعتبارسنجی استفاده شده است. شبکه با batch های با اندازه ۱۲۸ در ۱۵ epoch و بهینه ساز Adam با نرخ یادگیری 1×10^{-3} آموزش داده شده است، و به دقت ۹۹.۸٪ برای داده های آموزش و ۹۹.۵٪ درصد برای داده های اعتبارسنجی می رسد. پس از آموزش شبکه، مدل یادگیری شده ذخیره می شود تا در مرحله تست برخط استفاده شود. در این گام همانند بخش جمع آوری داده، اسپکتوگرام ها را از نمونه های ورودی تشکیل داده و بعد از پیش پردازش به مدل ذخیره شده به عنوان ورودی داده می شوند.

۵ چالش ها و کارهای آینده

چالش اصلی که در این پروژه وجود دارد جمع آوری و آماده سازی داده است. شبکه های عصبی عمیق برای یادگیری به داده زیاد نیاز دارند، و جمع آوری داده مورد نیاز در این پروژه ملزم به روشن و خاموش کردن متعدد وسایل می باشد که از نظر ایمنی وسایل چالش برانگیز

است. چالش بعدی این است که برای برخی از وسایل مانند هیتر و سشوار، اسپکتوگرام های مشابهی به دست می آیند که شبکه نمیتواند به خوبی آنها را از هم تمایز دهد. یک راه برای حل این مشکل این است که داده های بیشتری جمع آوری شوند که همان طور که گفته شد چالش برانگیز است. راه دیگری که میتوان ارائه داد، استفاده از Contrastive Learning است این روش بازنمایی ها را به نحوی یاد می گیرد که بازنمایی داده های یک کلاس به هم نزدیک شوند و بازنمایی داده های کلاس های متفاوت از یکدیگر دور شوند که باعث افزایش دقت در طبقه بندی می شود.