به نام خدا

دانشگاه تهران

دانشکده مهندسی برق و کامپیوتر

Trustworthy AI

تمرین شماره ۲

نام و نام خانوادگی	مهیار ملکی
شماره دانشجویی	۸۱۰۱۰۰۴۷۶
تاریخ ارسال گزارش	۱۴۰۲/۰۲/۳۰

فهرست گزارش سوالات

فهرست اشکال ۳

پرسش ۱ – SHAP ۴

الف ۴

۱. سه ویژگی منحصر به فرد SHAP ۴

۲. Kernel SHAP ۵

۳. Deep SHAP ۶

ب ۷

پیش‌پردازش داده‌ها ۷

مدل Regression ۷

Summary Plot ۸

Force Plot ۹

سوال ۲ – Knowledge Distillation ۱۱

مزایای مدل ۱۱

How it works? ۱۲

Loss Function ۱۲

Regularization ۱۳

سوال ۳ – D-RISE ۱۴

Summary ۱۴

Mask generation algorithm ۱۵

Similarity metric ۱۵

پیاده‌سازی ۱۶

سوال ۴ – LIME ۱۸

منابع ۲۱

فهرست اشکال

شکل ۱- عملیات forward و backward در شبکه‌های عصبی..... ۶
شکل ۲- نمودار تغییرات مقدار تابع هزینه حین آموزش برای دادگان آموزش و ارزیابی..... ۷
شکل ۳- نمودار summary مقادیر SHAP به دست آمده از روش kernelSHAP..... ۸
شکل ۴- نمودار summary مقادیر SHAP به دست آمده از روش deepSHAP..... ۹
شکل ۵- نمودار force مقادیر SHAP برای کشور رومانی با روش kernelSHAP..... ۱۰
شکل ۶- نمودار force مقادیر SHAP برای کشور لهستان با روش kernelSHAP..... ۱۰
شکل ۷- نمودار force مقادیر SHAP برای کشور رومانی با روش deepSHAP..... ۱۰
شکل ۸- نمودار force مقادیر SHAP برای کشور لهستان با روش deepSHAP..... ۱۰
شکل ۹- مراحل مدل D-RISE..... ۱۴
شکل ۱۰- تصویر چوب اسکی (تصویر چپ) - bounding box (تصویر وسط) - saliency map (تصویر راست)..... ۱۶
شکل ۱۱- تصویر دونات (تصویر چپ) - bounding box (تصویر وسط) - saliency map (تصویر راست)..... ۱۶
شکل ۱۲- تصویر عروسک خرسی (تصویر چپ) - bounding box (تصویر وسط) - saliency map (تصویر راست)..... ۱۷
شکل ۱۳- تصویر مار زنگی به همراه نواحی انتخاب شده و نقاط pros & cons و heatmap تاثیر نواحی تصویر..... ۱۸
شکل ۱۴- تصویر فانوس دریایی به همراه نواحی انتخاب شده و نقاط pros & cons و heatmap تاثیر نواحی تصویر..... ۱۹
شکل ۱۵- تصویر ادوات موسیقی به همراه نواحی انتخاب شده و نقاط pros & cons و heatmap تاثیر نواحی تصویر ۲۰

پرسش ۱ – SHAP

الف

۱. سه ویژگی منحصر به فرد SHAP

- Additive Feature Attribution Methods

در این روش‌ها، یک معیار اهمیت به هر ویژگی مدل یادگیری ماشین، بر اساس میزان مشارکت آن ویژگی در خروجی مدل نسبت می‌دهیم، تا در نهایت راهی برای توصیف رفتار مدل و یافتن تاثیرگزارترین ویژگی‌ها در پیش‌بینی مدل بیابیم. این روش‌ها مدلی تفسیری دارند که در واقع تابعی خطی از متغیرهای باینری می‌باشد:

$$g(z') = \phi_0 + \sum_{i=1}^{M} \phi_i z_i'$$

خصوصیت مهم این روش‌ها، وجود یک جواب یکتا با سه ویژگی منحصر فرد زیر می‌باشد:

۱. Local Accuracy

ویژگی دقت محلی بیان می‌کند که برای یک ورودی خاص x ، خروجی مدل تفسیری برای x' منطق بر خروجی مدل اصلی برای x می‌باشد. x' در واقع ورودی ساده شده‌ای برای x می‌باشد که با تابع مپینگ $x = h_x(x')$ به آن بازگردانده می‌شود.

$$f(x) = g(x') = \phi_0 + \sum_{i=1}^{M} \phi_i x_i'$$

۲. Missingness

این خصوصیت بیان می‌کند که اگر در فضای ویژگی‌ها، ویژگی‌ای وجود داشته باشد که در خروجی مدل بی‌تاثیر باشد، آنگاه وزن آن در مدل ساده شده، صفر خواهدبود.

$$x_i' = 0 \Rightarrow \phi_i' = 0$$

۳. Consistency

این ویژگی بیان می‌کند که اگر مدل به گونه‌ای تغییر کند که تاثیر برخی ورودی‌های ساده‌شده در مدل تفسیری، افزایش یافته یا ثابت بماند آنگاه تاثیر آن ورودی در مدل اصلی نباید کاهش بیابد.

$$f_x'(z') - f_x'(z' \backslash i) \geq f_x(z') - f_x(z' \backslash i) \quad \Rightarrow \quad \phi_i(f', x) \geq \phi(f, x)$$

۲. Kernel SHAP

این روش یک روش جایگزین برای تخمین مقادیر SHAP می‌باشد که از ایده روش LIME الهام گرفته‌است. روش KernelSHAP برای یک نمونه x اثر تک تک مقادیر ویژگی‌ها را در پیش‌بینی مدل تخمین می‌زند. ایده اصلی این روش، بر این اساس می‌باشد که به جای آموزش دوباره مدل بر زیرمجموعه‌ای از ویژگی‌ها، از خود مدل آموزش یافته f استفاده می‌کنیم و ویژگی‌های miss شده را با ویژگی‌های marginalize شده به دست آمده از نمونه‌ها، جایگزین می‌کنیم. به عنوان مثال برای سه ویژگی x_1 و x_2 و x_3، مدلی که ویژگی x_3 در آن miss شده باشد، به صورت زیر بدست می‌آید:

$$f_{\{x_1,x_2\}}(x_1,x_2) \xrightarrow[Kernel\ SHAP]{} \int f(x_1,x_2,x_3)p(x_3)dx_3$$

تخمین این مقدار همچنان نیازمند محاسبه $p(x_3)$ می‌باشد. برای انجام این کار، از روش LIME به همراه تابع مجاورت π استفاده می‌شود که بسیار متفاوت‌تر از LIME عمل می‌کند.

$$\pi_x^{\text{LIME}}(z) = \exp\left(-D(x,z)^2/\sigma^2\right) \qquad D: Distance\ Function$$

$$\pi_x^{\text{SHAP}}(z') = \frac{(p-1)}{\binom{p}{|z'|}|z'|(p-|z'|)} \qquad p: number\ of\ features$$

این دو معادله واضحا متفاوت می‌باشند، اولی از ویژگی‌های اصلی استفاده می‌کند و فواصل بین نقاط نمونه‌برداری شده و داده‌های اصلی که می‌خواهیم تفسیر کنیم را جریمه می‌کند. در حالی که در معادله دوم، کرنل SHAP تنها از تعداد ویژگی‌های موجود در زیرمجموعه $|z'|$ استفاده کرده و زیرمجموعه‌ها را با اختلاف تعداد ویژگی‌ها از مقدار 0 یا p جریمه می‌کند. این باعث می‌شود که وزن بیشتری برای زیرمجموعه‌هایی که تعداد کمی از ویژگی‌ها را شامل می‌شوند (رفتار مستقل ویژگی‌ها) یا تقریبا تمام آنها را شامل می‌شوند (تاثیر ویژگی‌ها در ارتباط با تمام ویژگی‌های دیگر)، در نظر گرفته شود.

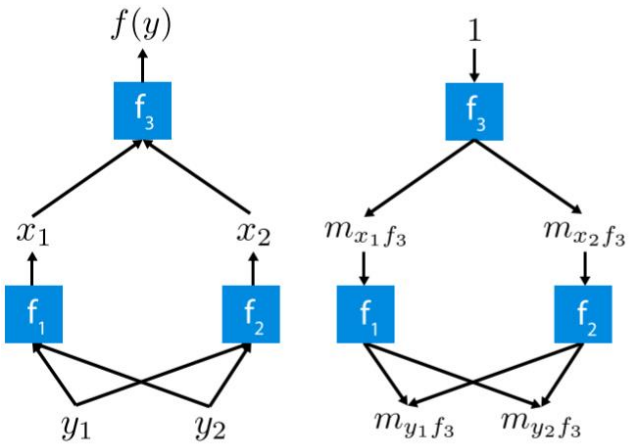
در واقع این روش از ۵ مرحله تشکیل شده‌است:

۱. انتخاب تعدادی نمونه ($z'_k \in \{0,1\}^M$) از داده‌ها
۲. انتقال نمونه‌های انتخاب شده z'_k به فضای ویژگی‌های اصلی و محاسبه پیش‌بینی مدل برای آن‌ها
۳. محاسبه وزن هر z'_k با استفاده از کرنل SHAP
۴. فیت کردن یک مدل خطی وزن‌دار
۵. وزن‌های مدل خطی ϕ^k به عنوان مقادیر SHAP در نظر گرفته می‌شوند

۳. Deep SHAP

روش KernelSHAP به عنوان یک روش مستقل از مدل برای تخمین مقادیر SHAP ، از منظر تعداد نمونه‌های مورد نیاز، کارایی تخمین را افزایش می‌دهد. حال با محدود کردن توجه به نوعی خاص از مدل‌ها (Deep Learning Models) می‌توان روشی سریعتر برای تخمین مقادیر SHAP توسعه داد. در واقع این سوال در اینجا مطرح می‌شود که آیا راهی وجود دارد تا با بهره گرفتن از دانش اضافه‌ای که نسبت به ماهیت مدل‌های عمیق می‌دانیم، بتوانیم کارایی محاسباتی را بهبود ببخشیم. جواب این سوال را با توجه به ارتباط مدل DeepLIFT و مقادیر SHAP می‌توان یافت. اگر ما مقدار مرجع در DeepLIFT را به عنوان نشان‌دهنده E[x] در مقادیر SHAP تفسیر کنیم، آنگاه DeepLIFT با فرض این که ویژگی‌های ورودی از هم مستقل‌اند و مدل شبکه عمیق خطی می‌باشد، مقادیر SHAP را تخمین می‌زند.

روش Deep SHAP مقدایر SHAP محاسبه شده برای اجزای کوچکتر شبکه را با مقادیر به دست آمده برای کل شبکه ترکیب می‌کند. این کار با عبور بازگشتی ضرب‌کننده‌های DeepLIFT ، که اینجا به عنوان مقادیر SHAP در نظر گرفته می‌شوند، به صورت عقب‌گرد از شبکه صورت می‌گیرد. (شکل ۱)



شکل ۱- عملیات **forward** و **backward** در شبکه‌های عصبی

ب

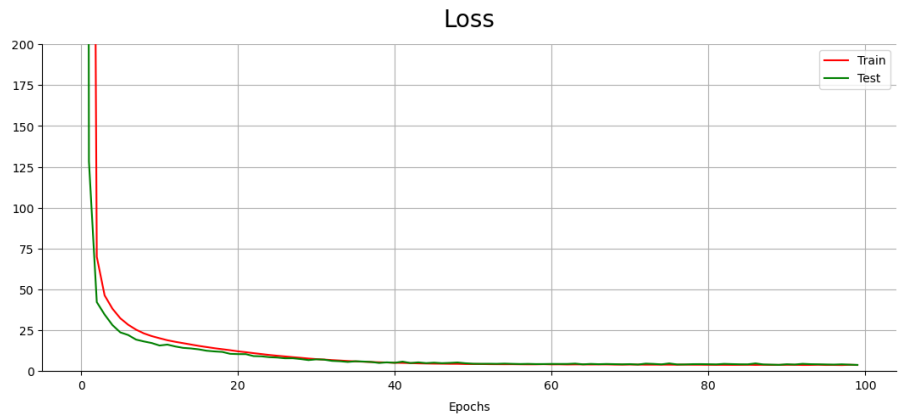
پیش‌پردازش داده‌ها

در این مرحله، کارهای انجام گرفته به شرح زیر می‌باشند:

۱. جایگزینی مقادیر صفر با جای خالی
۲. حذف سطرهایی که بیشتر از ۵ جای خالی دارند
۳. پر کردن جای خالی ها با مقدار میانه هر ستون
۴. حذف کشورهایی که تنها یک داده در جدول دارند
۵. انجام کدینگ one hot برای ستون‌های دسته‌ای
۶. نرمال کردن داده‌ها با استفاده از روش min max
۷. تقسیم داده‌ها به دو گروه آموزش و ارزیابی، به طوری که از هر کشور در هر دو گروه داده وجود داشته‌باشد. این کار شرط خواسته شده در صورت سوال را (وجود حداقل یک نمونه از ۳ کشور هر قاره در دادگان تست) تضمین می‌کند

مدل Regression

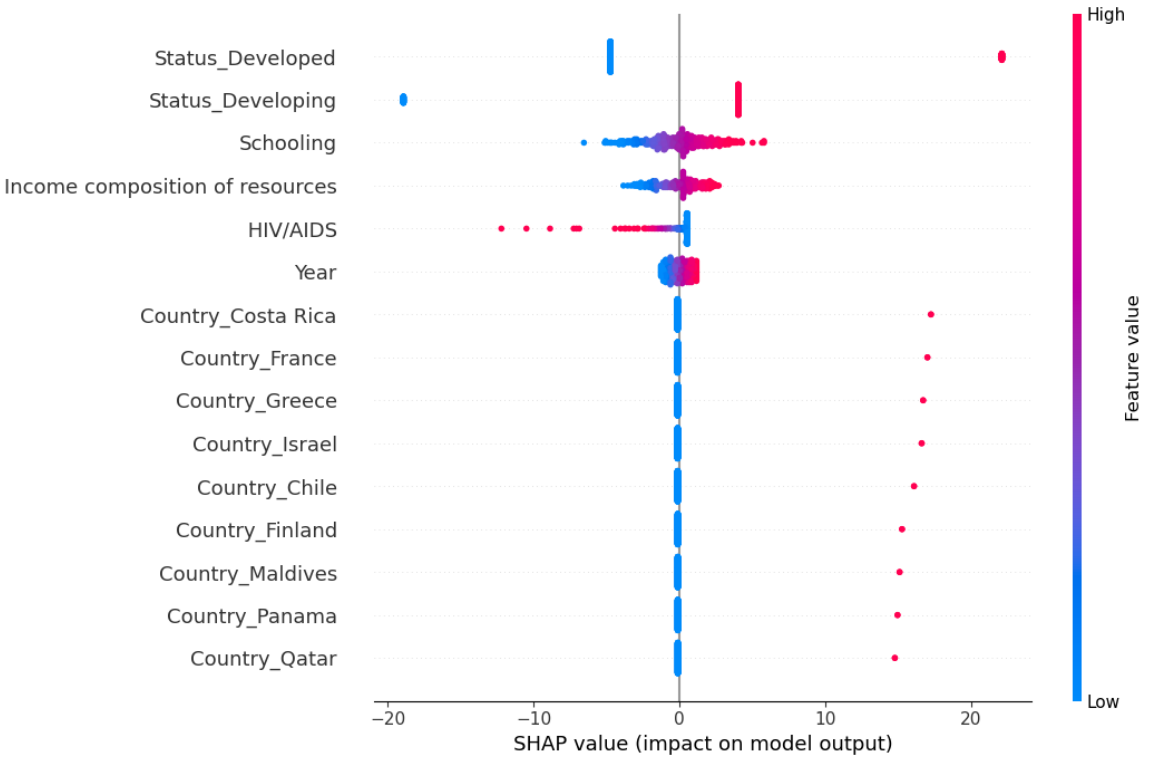
در این مرحله، یک شبکه عصبی ساده که دو لایه مخفی با تعداد نورون‌های ۱۲۸ و ۶۴ دارد را آموزش می‌دهیم. لازم به ذکر است که از تابع هزینه L2 و بهینه‌ساز Adam در اینجا استفاده کرده‌ایم. نمودار هزینه این مدل در حین آموزش در شکل ۲ قابل مشاهده است. چنان چه می‌بینیم مقدار تابع هزینه هم برای داده‌های آموزش و هم ارزیابی به خوبی کاهش یافته و همگرا شده‌است.



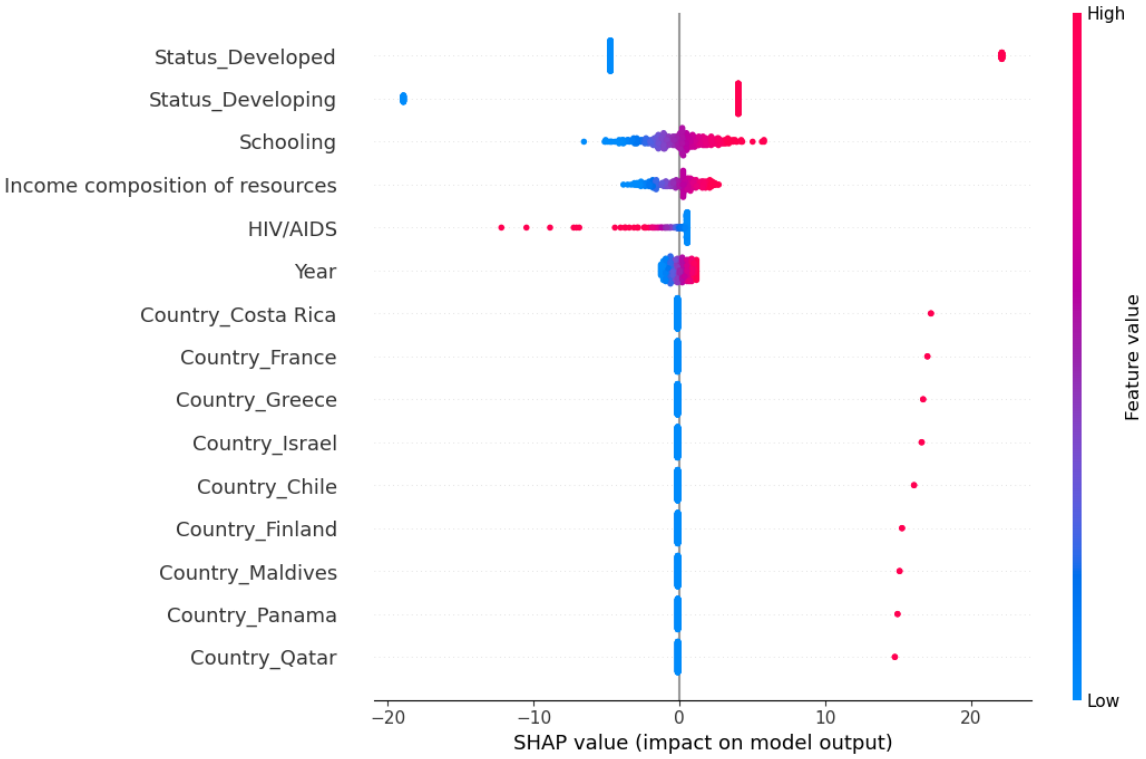
شکل ۲- نمودار تغییرات مقدار تابع هزینه حین آموزش برای دادگان آموزش و ارزیابی

Summary Plot

در این مرحله با اعمال دو روش Kernel SHAP و Deep SHAP به رسم نمودار Summary مقادیر SHAP برای هر دو روش مذکور و به روی داده‌های ارزیابی می‌پردازیم. نمودارهای مربوطه در شکل‌های ۳ و ۴ قابل مشاهده می‌باشند. همان طور که در این نمودارها مشخص است، مقادیر به دست آمده از هر دو روش، مشابهت زیادی به هم داشته و تا حد زیادی بر هم منطبق می‌باشند. با توجه به نمودارها مشاهده می‌کنیم که ویژگی‌های توسعه یافته بودن و در حال توسعه بودن، تاثیرگزارترین ویژگی‌ها بر سن امید به زندگی در بین کشورها می‌باشند. به عنوان مثال، توسعه یافته بودن یک کشور، تاثیر زیاد و مثبتی داشته و باعث افزایش سن امید به زندگی مردم می‌شود، در حالی که در حال توسعه بودن یک کشور، تاثیری زیاد ولی منفی خواهد داشت، لذا باعث کاهش سن امید به زندگی خواهدشد. در رتبه‌های بعدی از منظر تاثیرگزاری، ویژگی‌هایی مانند درآمد از منابع یا تعداد سال‌های تحصیل قرار دارند.



شکل ۳- نمودار **summary** مقادیر **SHAP** به دست آمده از روش **kernelSHAP**



شکل ۴- نمودار **summary** مقادیر **SHAP** به دست آمده از روش **deepSHAP**

Force Plot

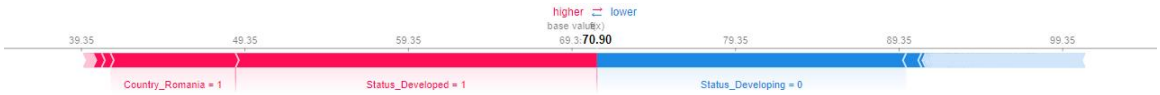
در این مرحله به رسم و بررسی نمودار Force بر اساس مقادیر SHAP به دست آمده با روش‌های Kernel و Deep برای دو کشور دلخواه رومانی و لهستان از قاره اروپا می‌پردازیم. مشخصات نمودار force:

- F(x) : پیش‌بینی مدل
- Base value : میانگین پیش‌بینی مدل روی داده‌های تست
- رنگ آبی : ویژگی‌هایی که اثر منفی دارند
- رنگ قرمز : ویژگی‌هایی که اثر مثبت دارند

با توجه به نمودار مربوط به کشور رومانی در شکل‌های ۵ و ۷ قابل مشاهده است، پیش‌بینی مدل نزدیک به میانگین پیش‌بینی مدل برای دادگان ارزیابی می‌باشد. همچنین در حال توسعه نبودن بیشترین تاثیر منفی و پیشرفته بودن بیشترین تاثیر مثبت را دارد. همچنین بودن در کشور رومانی خود به تنهایی تاثیر زیادی در افزایش سن امید به زندگی دارد.

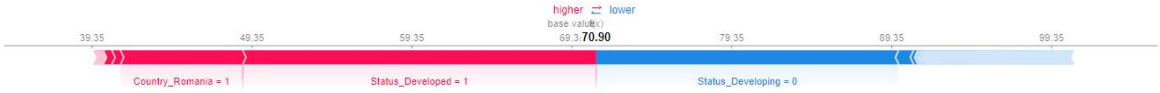
در مورد کشور لهستان نیز که در شکل‌های ۶ و ۸ قابل مشاهده است، پیش‌بینی مدل بیشتر از میانگین سن امید به زندگی برای دادگان ارزیابی می‌باشد. همچنین مانند کشور رومانی، در حال توسعه نبودن بیشترین تاثیر منفی و پیشرفته بودن بیشترین تاثیر مثبت را داشته و بودن در کشور لهستان خود به تنهایی تاثیر زیادی در افزایش سن امید به زندگی دارد. پس از این ویژگی‌ها تعداد سال‌های تحصیلی و میزان درآمد از منابع تاثیرگزارترین ویژگی‌ها هستند.

لازم به ذکر است که مقادیر به دست آمده از هر دو روش kernel و deep کاملا برابر بوده و بر هم منطق می‌باشند.

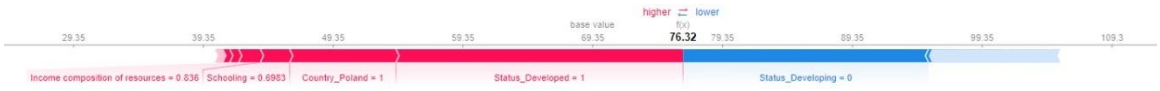


شکل ۵- نمودار **force** مقادیر **SHAP** برای کشور رومانی با روش **kernelSHAP**

شکل ۶- نمودار **force** مقادیر **SHAP** برای کشور لهستان با روش **kernelSHAP**



شکل ۷- نمودار **force** مقادیر **SHAP** برای کشور رومانی با روش **deepSHAP**



شکل ۸- نمودار **force** مقادیر **SHAP** برای کشور لهستان با روش **deepSHAP**

سوال ۲ – Knowledge Distillation

مزایای مدل

ثابت شده‌است که شبکه‌های عصبی عمیق برای تسک‌های طبقه‌بندی بسیار موثر می‌باشند، به خصوص در زمانی که ابعاد داده‌ها بزرگ باشد یا رابطه بین ورودی و خروجی پیچیده باشد یا تعداد نمونه‌های آموزش زیاد باشد. اما تفسیر این که چرا یک شبکه عصبی عمیق برای یک نمونه خاص، یک تصمیم مشخص را اتخاذ می‌کند، امری دشوار می‌باشد. اگر بازنمایی مدل مشابه یک درخت تصمیم بود که می‌توانستیم یک سری متغیرها را برای آن در نظر بگیریم و یک مسیر برای هر تصمیم بر اساس این متغیرها ارائه دهیم، تفسیر نحوه پیش‌بینی این مدل‌ها امری ساده بود. مسئله‌ای که اینجا مطرح است، trade-off بین دقت و تعمیم‌پذیری مدل‌ها می‌باشد. درخت‌های تصمیمی که دقت خوبی دارند، تعمیم‌پذیری خوبی نخواهند داشت و بالعکس. در واقع هدف مقاله، این می‌باشد که نحوه عملکرد شبکه را به شکل یک درخت تصمیم تفسیر کند و تلاش می‌کند تا وضعیت trade-off گفته‌شده برای درخت‌های تصمیم را بهبود ببخشد.

در نهایت مقاله مدلی را معرفی می‌کند که از مزایای زیر برخوردار می‌باشد:

- تفسیرپذیری: از آنجایی که مدل بر اساس درخت تصمیم می‌باشد، تعقیب تصمیم‌گیری‌های منتهی به یک خروجی خاص ساده‌تر است.
- پیچیدگی محاسباتی: تقطیر یک مدل سنگین به یک مدل سبکتر، باعث شده‌است تا در عین حفظ تقریبی دقت مدل، پیچیدگی محاسباتی کاهش بیابد.

How it works?

اولین جزء درخت تصمیم، گره‌های تصمیم هستند. در این مدل، چیزی که برای هر گره نیاز داریم، مقادیر احتمالاتی بر اساس داده ورودی می‌باشد. برای بدست آوردن آن از وزن‌ها و فعال‌سازها شبکه استفاده می‌کنیم. در هر گره ابتدا ترکیب خطی متغیرهای ورودی را گرفته و تابع سیگموید را روی مجموع آن‌ها اعمال می‌کنیم تا در نهایت احتمال انشعاب را بدست آوریم (p_i). هر گره شامل یک بردار n بعدی می‌باشد که n تعداد کلاس‌هاست. این بردار نشان‌دهنده توزیع احتمالی نمونه‌ها در یک کلاس می‌باشد.

$$p_i(\mathbf{x}) = \sigma(\mathbf{x}\mathbf{w}_i + b_i)$$

همانند یک درخت تصمیم نرم، خروجی این درخت نشان‌دهنده توزیع احتمالی کلاس‌ها می‌باشد. توزیع خروجی برابر با مجموع توزیع‌ها ضرب در احتمال مسیر رسیدن به آن توزیع، می‌باشد(Q).

$$Q_k^\ell = \frac{\exp\left(\phi_k^\ell\right)}{\sum_{k'} \exp\left(\phi_{k'}^\ell\right)}$$

در واقع، این مدل با استفاده از توزیع برگی که بیشترین احتمال مسیر را دارد، یک توزیع پیش‌بینی برای کلاس‌ها ارائه می‌دهد.

Loss Function

این مدل از تابع هزینه‌ای مشابه cross-entropy استفاده می‌کند. در واقع این تابع تلاش می‌کند تا هزینه cross-entropy بین هر برگ و توزیع خروجی‌اش را کمینه کند. لازیم به ذکر است که برای محاسبه هزینه cross-entropy ، هر برگ با احتمال مسیرش وزن‌دهی می‌شود. فرمولاسیون این تابع هزینه به صورت زیر می‌باشد:

$$\mathrm{L}(\mathbf{x}) = -\log\left(\sum_{\ell \in \text{ Leaf Nodes}} P^\ell(\mathbf{x}) \sum_k T_k \log\, Q_k^\ell\right)$$

Regularization

یک مشکل بزرگ بزرگ الگوریتم درخت تصمیم، گیر افتادن در برخی از نقاط بهینه محلی ضعیف می‌باشد. به همین خاطر به یک ترم regularization نیاز داریم تا انشعاب نامتوازن را جریمه کند و انشعابی را ترجیح دهد که به طور مساوی از هر زیردرخت چپ یا راست استفاده کند.

این جریمه یک cross-entropy بین میانگین توزیع مورد نظر (مساوی از هر زیردرخت چپ یا راست) و میانگین توزیع واقعی می‌باشد:

$$\alpha_i = \frac{\sum_x P^i(x)\, p_i(x)}{\sum_x P^i(x)}$$

در این معادله، $P^i(x)$ نشان‌دهنده احتمال مسیر از ریشه تا گره i برای یک نمونه x می‌باشد. سپس مجموع جریمه‌ها برای تمام گره‌های داخلی به شکل زیر محاسبه می‌شود:

$$C = -\lambda \sum_{i \in \text{ Inner Nodes}} 0.5\log(\alpha_i) + 0.5\log(1-\alpha_i)$$

لاندا در این معادله نشان‌دهنده هایپرپارامتری می‌باشد که میزان تاثیر جریمه را تنظیم می‌کند.

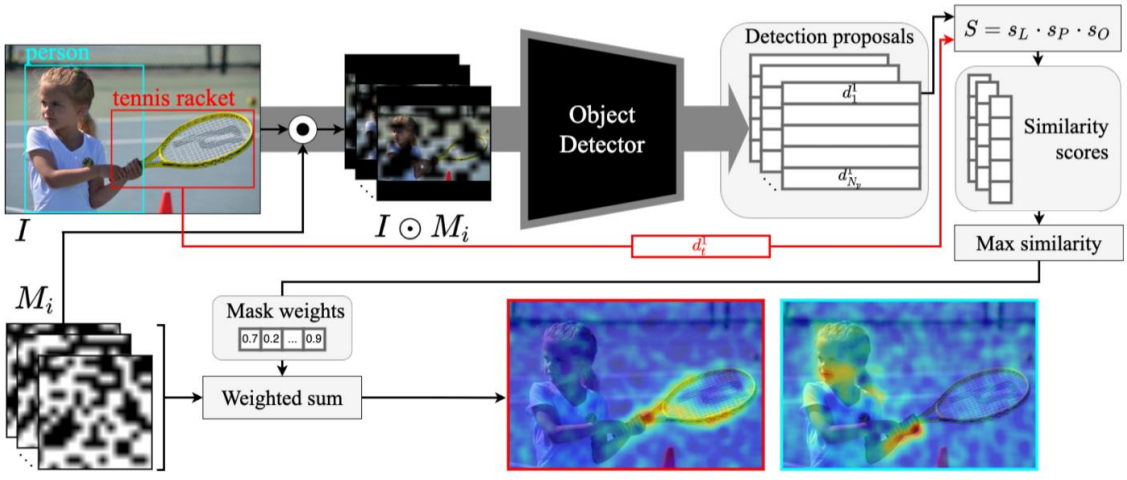
سوال ۳ – D-RISE

Summary

Saliency map ها ابزار محبوبی برای بررسی و تفسیر شبکه‌های عصبی می‌باشند. آن‌ها ناحیه‌هایی از داده ورودی که اهمیت بیشتری در پیش‌بینی مدل دارند را مشخص می‌کنند. اکثر این روش‌ها بر تسک طبقه‌بندی تمرکز داشته‌اند، اما در این مقاله به تسک کمتر کار شده‌ی شناسایی اشیاء پرداخته شده است.

نواحی تاثیرگزار برای شناسایی یک موجودیت در تصویر، ممکن است که با خود آن موجودیت منطبق نباشد. به عنوان مثال ممکن است مدل از اطلاعات موجود در زمینه استفاده کند یا در موارد دیگر ممکن است بخش‌های مختلف یک موجودیت از نظر اهمیت متفاوت باشند. الگوریتم‌های طبقه‌بندی موجود، در شناسایی این نواحی ضعف دارند.

بنابراین این سوال مطرح می‌باشد که در یک تسک شناسایی اشیاء، تاثیر نواحی مختلف تصویر در عملیات شناسایی به چه شکل است؟ نویسندگان این مقاله، با بررسی چگونگی اثر perturbationهای مختلف نمونه ورودی بر خروجی مدل، به پاسخ به این سوال پرداخته‌اند. آن‌ها تصاویر ماسک شده را از مدل مورد نظر گزرانده‌اند و برای هر کدام یک بردار شناسایی proposal به دست آمده‌است. برای مقایسه این بردارهای شناسایی proposal با بردار مورد نظر برای تفسیر، یک معیار شباهت ارائه داده‌اند. با استفاده از این معیار، مقدار شباهت برای هر ماسک محاسبه می‌شود و سپس saliency map مورد نظر، با جمع وزن‌دار آن‌ها به دست می‌آید. در نتیجه تنها زمانی که ماسک، نواحی تاثیرگزار را شامل شود، موجودیت مورد نظر همچنان شناسایی شده و امتیاز بالایی خواهدداشت، لذا در saliency map نواحی تاثیرگزار، مقدار بیشتری خواهنددداشت. (شکل ۹)



شکل ۹- مراحل مدل **D-RISE**

Mask generation algorithm

الگوریتم تولید ماسک در این مقاله از مدل RISE برگرفته شده و دارای مراحل زیر می‌باشد:

۱. N نمونه‌ی باینری با سایز $h \times w$ که کوچکتر از تصویر اصلی می‌باشند($H \times W$)، با قرار دادن مقدار ۱ برای هر پیکسل با احتمال p و صفر با احتمال 1-p ، ایجاد می‌کنیم.

۲. با استفاده از interpolation دوتایی تمامی ماسک‌ها را upsample می‌کنیم به طوری که اندازه آن‌ها $(h+1)C_H \times (w+1)C_W$ شود.

سایز هر سلول در ماسک upsample شده $C_H \times C_W = [H/h] \times [W/w]$ می‌باشد.

۳. نواحی‌ای را با مقداردهی رندوم یکنواخت از بازه (0,0) تا (C_H,C_W) کراپ می‌کنیم.

Similarity metric

هر دو بردار proposal و بردار هدف از سه بخش تشکیل می‌شوند:

- Localization : مختصات bounding box ها
- Classification : احتمال کلاس‌ها
- Objectless score : مقدار عددی بین صفر و یک

معیار شباهت با اندازه‌گیری مشابهت بین هر کدام از این بخش‌ها به صورت جداگانه، محاسبه می‌شود.

برای محاسبه شباهت بین دو بردار از منظر localization از معیار IoU بین Bounding boxها استفاده می‌شود:

$$s_L(d_t, d_j) = \text{IoU}\left(L_t, L_j\right)$$

برای محاسبه شباهت بین دو بردار از منظر طبقه‌بندی از معیار شباهت کسینوسی بین توزیع کلاس‌ها استفاده می‌شود:

$$s_P(d_t, d_j) = \frac{P_t \cdot P_j}{\|P_t\|\|P_j\|}$$

و برای محاسبه شباهت بین دو بردار از منظر objectness score از مقادیر objectness استفاده می‌شود:

$$s_O(d_t, d_j) = O_j$$

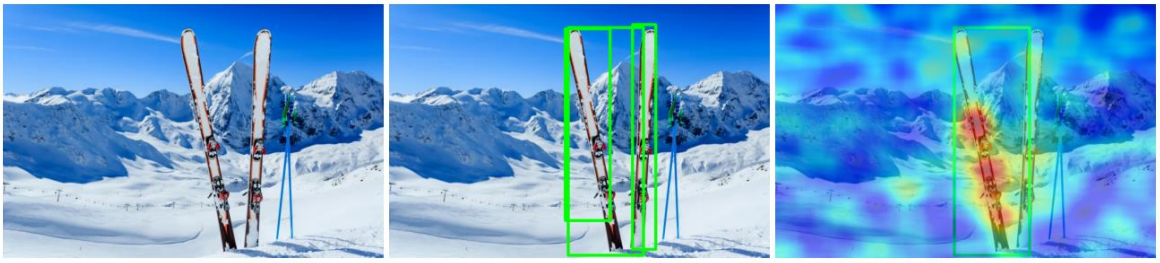
در نهایت سه مقدار به دست آمده در هم ضرب شده و معیار شباهت برای دو بردار به دست می‌آید. به دلیل خاصیت عملیات ضرب، اگر هر یک از مقادیر به صفر نزدیک باشند، معیار شباهت هم به صفر نزدیک خواهدشد.

$$s(d_t, d_j) = s_L(d_t, d_j) \cdot s_P(d_t, d_j) \cdot s_O(d_t, d_j)$$

پیاده‌سازی

در این قسمت برای به دست آوردن saliency mapها از مقدار احتمال 0.3 استفاده شده‌است.

در شکل ۱۰، چوب اسکی انتخاب شده در تصویر سوم با احتمال ۹۳ درصد شناسایی شده‌است. همچنین نقاط تاثیرگزار در این انتخاب، چنان چه قابل مشاهده است، بر نواحی مرکزی چوب اسکی که محل قرارگیری پاها می‌باشند، تمرکز دارد.

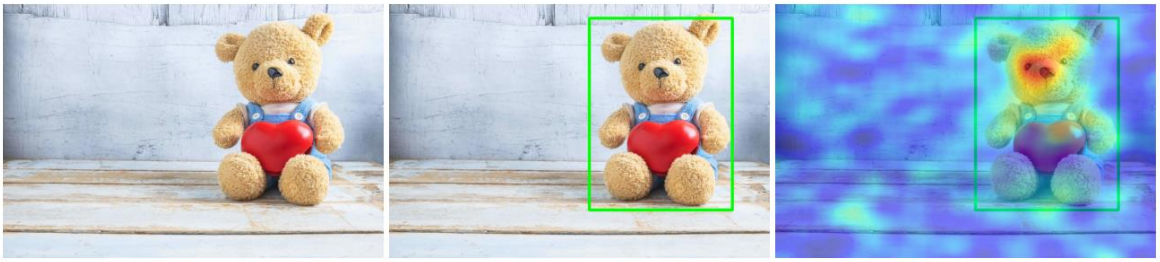


شکل ۱۰- تصویر چوب اسکی (تصویر چپ) - **bounding box** (تصویر وسط) - **saliency map** (تصویر راست)

در شکل ۱۱ نیز دونات موجود در مرکز تصویر با احتمال ۹۳ درصد انتخاب شده‌است. عامل تاثیرگزار در این انتخاب به نظر می‌رسد شکل حلقوی دونات و رنگ آن باشد. لازم به ذکر است که دونات‌های دیگر موجود در تصویر با احتمال‌های بیشتر نزدیک به ۱۰۰ درصد انتخاب شده‌اند و این کاهش احتمال ممکن است به خاطر شکلات‌های رنگی روی دونات باشد.

شکل ۱۱- تصویر دونات (تصویر چپ) - **bounding box** (تصویر وسط) - **saliency map** (تصویر راست)

در شکل ۱۲ نیز عروسک خرسی با احتمال نزدیک به ۱۰۰ درصد انتخاب شده‌است. نواحی تاثیرگزار در این انتخاب نیز در صورت و به خصوص پوزه و چشم‌ها متمرکز می‌باشد.



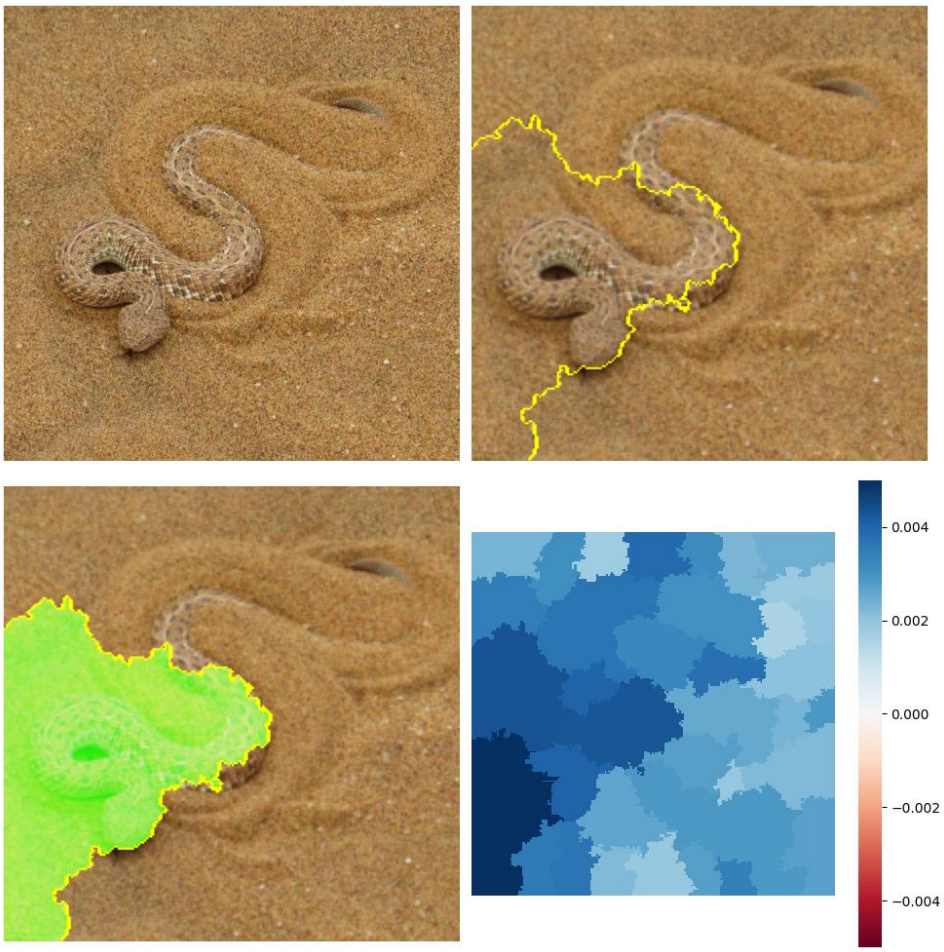
شکل ۱۲- تصویر عروسک خرسی (تصویر چپ) - **bounding box** (تصویر وسط) - **saliency map** (تصویر راست)

سوال ۴ – LIME

در این بخش، تصویر یک مار زنگی انتخاب شده‌است. احتمالات به دست آمده شبکه به صورت زیر می‌باشد:

```
(0.6478405, 'horned viper')
(0.32613173, 'sidewinder')
(0.013589431, 'leatherback turtle')
(0.0027936043, 'knot')
(0.0018535309, 'sea snake')
```

چنان چه مشخص است، بیشترین احتمال به دست آمده مربوط به کلاس افعی شاخ‌دار می‌باشد که اشتباه است (البته مار بودن تصویر به درستی تشخیص داده شده و تنها نوع آن اشتباه است) همچنین احتمال به دست آمده برای کلاس درست ۳۲ درصد می‌باشد. نکته‌ای که در شکل ۱۳ قابل مشاهده است، این است که نواحی مربوط به سر و پیچش بدن مار تاثیر مثبت زیادی در پیش‌بینی مدل داشته‌است.

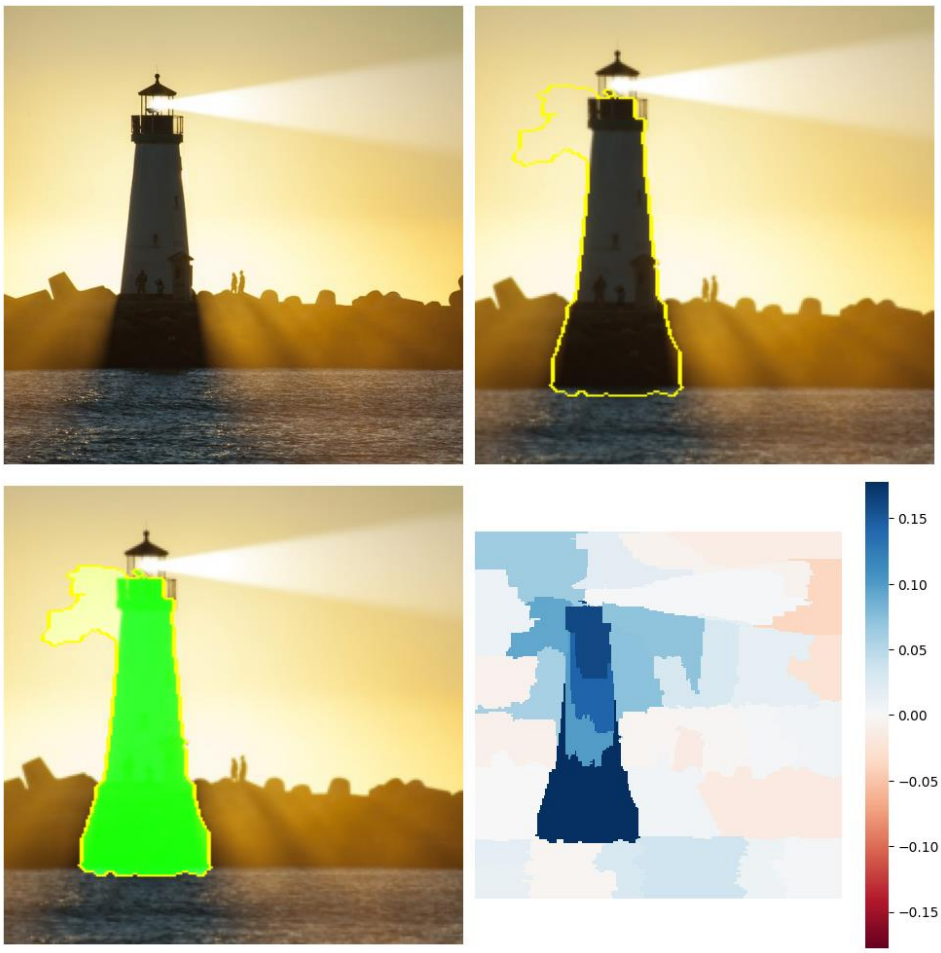


شکل ۱۳- تصویر مار زنگی به همراه نواحی انتخاب شده و نقاط **pros & cons** و **heatmap** تاثیر نواحی تصویر

در این قسمت، تصویر یک فانوس دریایی انتخاب شده‌است. احتمالات به دست آمده از شبکه به صورت زیر می‌باشد:

```
(0.8839334, 'beacon')
(0.082126535, 'breakwater')
(0.010992705, 'drilling platform')
(0.0069807284, 'container ship')
(0.0052685714, 'submarine')
```

چنان چه مشخص است، کلاس درست با احتمال بالای ۸۸ درصد انتخاب شده‌است. نکته‌ای که در شکل ۱۴ قابل مشاهده است، این است که نواحی مربوط به بدنه مخروطی فانوس، تاثیر مثبت زیادی در پیش‌بینی داشته‌است.

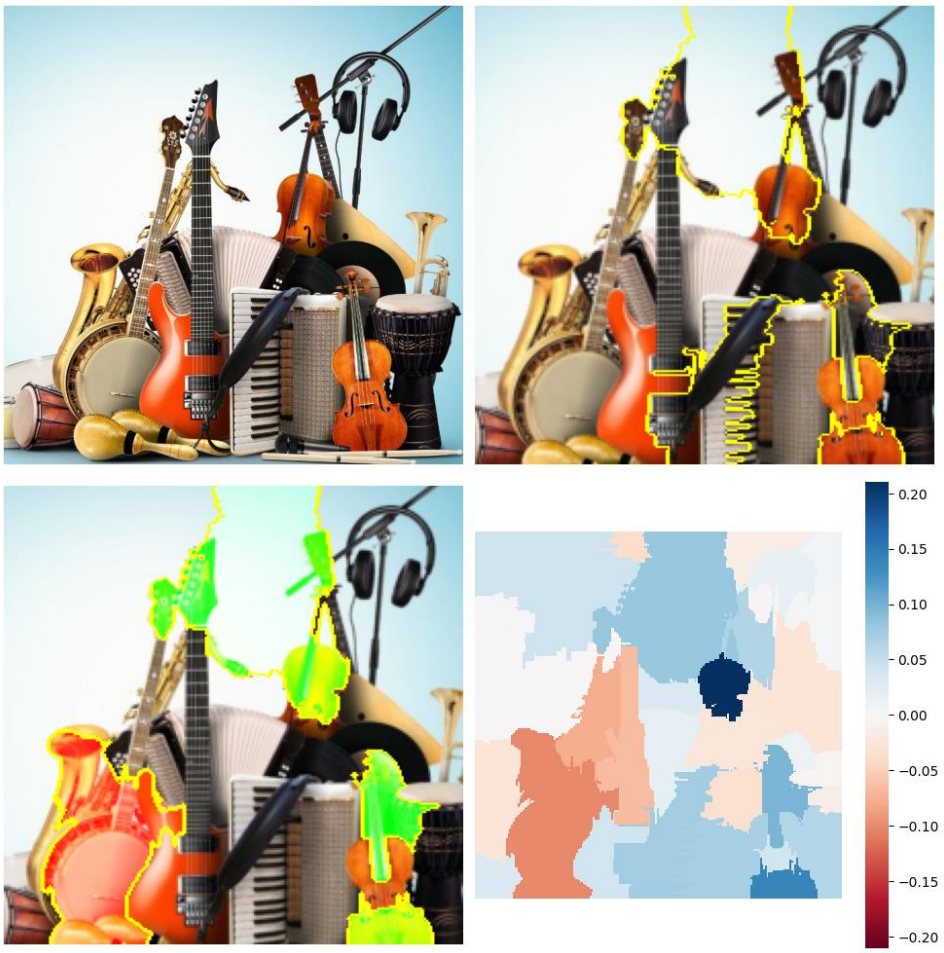


شکل ۱۴- تصویر فانوس دریایی به همراه نواحی انتخاب شده و نقاط **pros & cons** و **heatmap** تاثیر نواحی تصویر

در این قسمت، همان طور که در صورت سوال خواسته‌شده، تصویری انتخاب شده‌است که چند کلاس در آن وجود داشته‌باشد. احتمالات به دست آمده از شبکه به صورت زیر می‌باشد:

```
(0.7117781, 'cello')
(0.19297945, 'violin')
(0.015977144, 'stage')
(0.013588687, 'acoustic guitar')
(0.01138448, 'banjo')
```

چنان چه مشخص است، با وجود چندین نوع ساز مختلف در تصویر، اما بیشترین احتمال خروجی مدل با ۷۱ درصد مربوط به کلاس ویولونسل می‌باشد. همچنین با توجه به شکل ۱۵ می‌بینیم که نتیجه به دست آمده تایید می‌شود و نواحی مربوط به ویولونسل تاثیر مثبت زیادی در خروجی مدل داشته‌است.



شکل ۱۵- تصویر ادوات موسیقی به همراه نواحی انتخاب شده و نقاط **pros & cons** و **heatmap** تاثیر نواحی تصویر

منابع

[1] https://arxiv.org/pdf/1705.07874.pdf

[2] https://arxiv.org/pdf/1711.09784.pdf

[3] https://arxiv.org/pdf/2006.03204.pdf

[4] https://arxiv.org/pdf/1602.04938.pdf

[5] https://www.linkedin.com/pulse/make-neural-network-more-explainable-soft-decision-tree-angela-ju/

[6] https://www.youtube.com/watch?v=AW063Nju9F4

[7] https://github.com/marcotcr/lime/blob/master/doc/notebooks/Tutorial%20-%20images%20-%20Pytorch.ipynb

[8] https://christophm.github.io/

[9] https://data4thought.com/kernel_shap.html

[10] https://medium.com/razorthink-ai/distilling-a-neural-network-into-a-soft-decision-tree-1d1818dc1c4f

[11] https://towardsdatascience.com/explain-any-models-with-the-shap-values-use-the-kernelexplainer-79de9464897a