به نام خدا

دانشگاه تهران

دانشکده مهندسی برق و کامپیوتر

Trustworthy AI

تمرین شماره ۲

نام و نام خانوادگی	ثمین مهدی زاده
شماره دانشجویی	۸۱۰۱۰۰۵۲۶
تاریخ ارسال گزارش	۱۴۰۲-۰۲-۲۶

فهرست گزارش سوالات (لطفاً پس از تکمیل گزارش، این فهرست را به‌روز کنید.)

سوال 1 – SHAP

در این سوال سعی شده است تا به کمک داده های Life Expectency و متد های Deep SHAP و Kernel SHAP عملکرد یک مدل رگرسیون بررسی و تحلیل شود.

الف)

۱. روش های additive feature attribution داری یک مدل explanation هستند که یک تابع خطی از متغیر های باینری است:

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

که در آن z'_i آرایه ای از صفر و یک ها(حضور با عدم حضور ویژگی)، M تعداد ویژگی ها و Φ_i یک عدد حقیقی است. در واقع روش هایی که از این تابع خطی استفاده می کنند برای هر ویژگی یک وزن در نظر میگیرند و با جمع تاثیر تمام ویژگی ها بر روی ورودی های مختلف خروجی مدل اصلی را تخمین می زنند.

Local accuracy

این ویژگی بیان می کند در صورتی که ورودی اصلی را مقدار کمی ساده کنیم(x') و سپس به کمک این وروی و با استفاده از explanation model یک خروجی تولید کنیم خروجی تولید شده با خروجی مدل اصلی هنگام دادن وروی اصلی(x) باید مطابقت داشته باشد.

$$f(x) = g(x') = \phi_0 + \sum_{i=1}^{M} \phi_i x'_i \qquad (2)$$

Missingness

این خصیصه به این معنی است که در صورتی که یک ویژگی در ورودی اصلی نباشد نباید تاثیری بر روی خروجی مدل explanation داشته باشد.

$$x'_i = 0 \implies \phi_i = 0 \qquad (3)$$

Consistancy

اگر از وروی ساده شده برای پیش بینی استفاده شود و مشارکت یک ویژگی خاص برای به دست آمدن آن خروجی افزایش یابد یا تغییری نکند. وزن اختصاص داده شده برای آن ویژگی نباید کاهش بیابد.

$$f'_x(z') - f'_x(z' \setminus i) \geq f_x(z') - f_x(z' \setminus i) \qquad (4)$$

for all inputs $z' \in \{0, 1\}^M$ *, then* $\varphi_i(f', x) \geq \varphi_i(f, x)$.

۲. روش SHAP مقادیر shapley values را با در نظرگرفتن تمام جایگشت های ممکن برای ویژگی ها محاسبه می کند، در حالی که در روش kernel SHAP میزان مشارکت هر ویژگی با در نظر گرفتن یک زیر مجموعه تصادفی از ویژگی ها و وزن دهی به هر کدام از نمونه ها با استفاده از یک kernel function است. این kernel به این صورت عمل می کند که به نمونه های با تعداد فیچر خیلی کم یا خیلی زیاد امتیاز بیشتری اختصاص می دهد چرا که ممکن است حاوی اطلاعات بیشتری برای یک ویژگی خاص باشند. در واقع در این روش مدل خطی explanation را به وسیله loss زیر آموزش میدهیم.

$$\pi_{x'}(z') = \frac{(M-1)}{(M \; choose \; |z'|)|z'|(M-|z'|)}, \tag{5}$$

$$L(f, g, \pi_{x'}) = \sum_{z' \in Z} \left[f(h_x^{-1}(z')) - g(z') \right]^2 \pi_{x'}(z'), \tag{6}$$

where |z | *is the number of non-zero elements in* z

در فرمول بالا تابع Π همان کرنل ماست که امتیاز هر نمونه توسط آن مشخص می شود. تصویر زیر مراحل روش kernel shap را به خوبی توضیح میدهد.

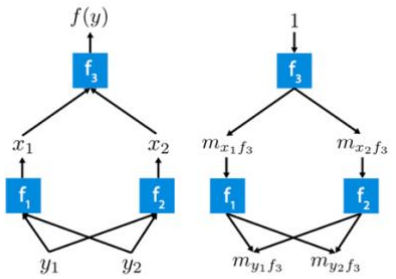
KernelSHAP estimates for an instance x the contributions of each feature value to the prediction.
KernelSHAP consists of five steps:

- Sample coalitions $z'_k \in \{0,1\}^M, \quad k \in \{1, \ldots, K\}$ (1 = feature present in coalition, 0 = feature absent).
- Get prediction for each z'_k by first converting z'_k to the original feature space and then applying model $\hat{f} : \hat{f}(h_x(z'_k))$
- Compute the weight for each z'_k with the SHAP kernel.
- Fit weighted linear model.
- Return Shapley values ϕ_k, the coefficients from the linear model.

شکل ۱. الگوریتم kernel shap

۳. مدل Deep Shap در واقع از ترکیب دو روش shapley values و DeepLift استفاده می کند.در مدل DeepLift به این صورت عمل می شود که یک ورودی به عنوان رفرنس انتخاب می شود و سپس تاثیر هروروی به این صورت در نظر گرفته می شود که اختلاف خروجی این ورودی با مقدار رفرنس در لایه های پایینی شبکه تقسیم می شود و به صورت بازگشتی تا اولین لایه شبکه تاثیر هر ویژگی به دست می آید. در واقع این روش فرض می کند که ویژگی های ورودی از یکدیگر مستقل هستند و بخش های غیر خطی شبکه را به صورت خطی در نظر میگیرد تا در نهایت تاثیر هر ویژگی را بر خروجی مدل به دست آورد. روش deep shap برای محاسبه ی shaply values نیز از یک روش مشابه استفاده می کند به این صورت که برای به دست آوردن مقادیر SHAP برای کل شبکه مقادیر SHAP برای هر بخش را محاسبه می کند و به صورت بازگشتی این مقادیر را تا رسیدن به اولین لایه شبکه محاسبه می کند. طبیعتا این روش از ساختار یک شبکه عصبی برای محاسبه مقادیر SHAP استفاده می کند و برخلاف kernel shap قابل اجرا بر روی هر مدلی نمی باشد. در واقع در روش deep shap مقادیر shapley values در یک پروسه سلسه مراتبی محاسبه می شوند

و همین موضوع باعث می شود از نظر محاسباتی به صرفه تر باشد در حالی که در روش kernel shap از یک تابع برای وزن دهی به نمونه ها و محاسبه این مقادیر استفاده می شود.



شکل۲. به دست آمدن مقادیر SHAP در الگوریتم deep shap

$$m_{y_i f_3} = \sum_{j=1}^{2} m_{y_i f_j} m_{x_j f_3} \qquad \text{chain rule} \qquad (7)$$

$$\phi_i(f_3, y) \approx m_{y_i f_3}(y_i - E[y_i]) \qquad \text{linear approximation} \qquad (8)$$

ب)

در این بخش با استفاده از داده های life expectancy و به کمک یک مدل رگرسیون ساده یک پیش بینی برای سن امید به زندگی افراد انجام شده است و سپس مدل به دست آمده تفسیر می شود.

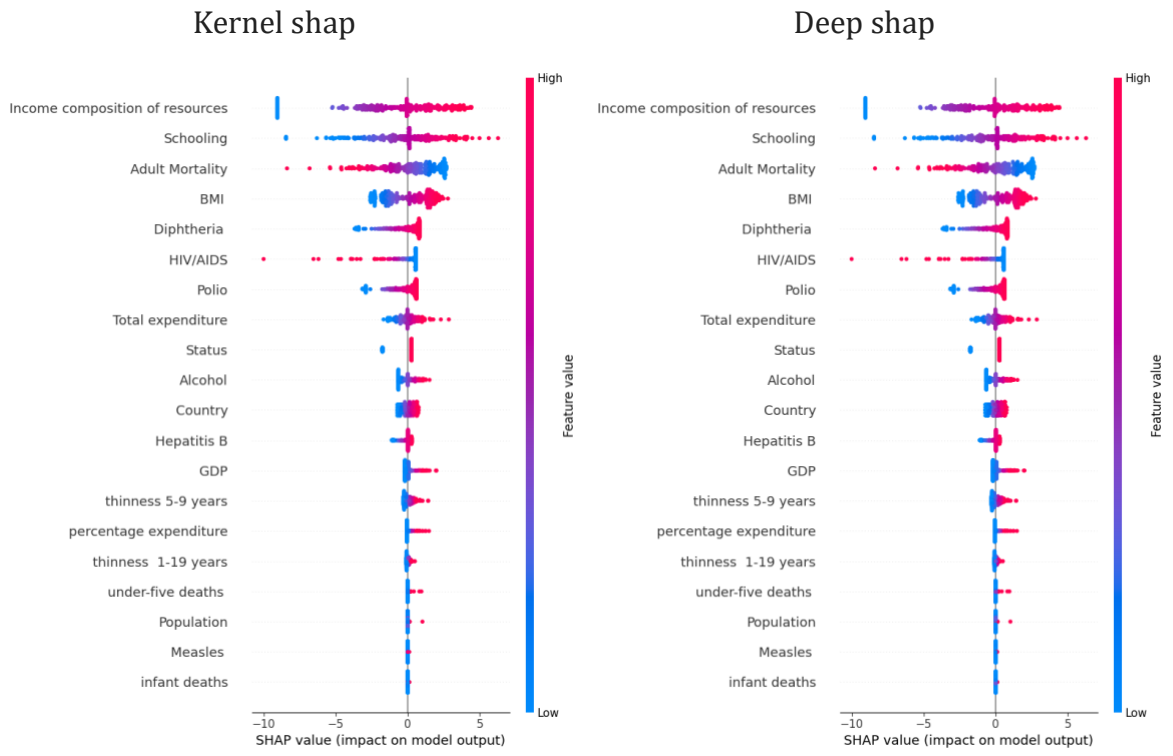
در ابتدا ۱۰ درصد از داده ها را برای تست جدا کرده و همان طور که مشاهده می شود در داده های تست در هر قاره حتما از ۳ کشور حداقل یک نمونه وجود دارد:

Country Count
Asia : 37
Europe : 32
Africa : 44
North America : 18
South America : 10
Oceania : 8

سپس پیش پردازش هایی جهت تمیز سازی داده ها مانند پر کردن نمونه های خالی، حذف برخی نمونه ها و یا تبدیل متغیر های categorical انجام می شود و به کمک یک مدل با دو لایه پنهان پیش بینی بر روی داده های تست انجام می شود. تابع loss در نظر گرفته شده برای این مدل mse و تابع بهینه ساز SGD می باشد.

پس از آموزش مدل مقدار RMSE بر روی داده های تست به ۴.۶۸رسیده است و مقدار r2square نیز برابر با ۰.۷۶ است که نشان می دهد تقریبا به مدل مطلوبی رسیده ایم.

در ادامه و برای تفسیر مدل با دو روش kernel shap و deep shap مقادیر shap برای تمام نمونه های تست به کمک summary plot رسم شده اند.



شکل۳. نمودار summary plot برای دو الگوریتم معرفی شده

همان طور که مشاهده می شود نتایج هر دو روش با هم مطابقت دارند. در این تصویر می توان اهمیت ویژگی ها را مشاهده کرد. هر چه از بالا به سمت پایین می رویم اهمیت ویژگی ها برای تصمیم گیری مدل کاهش می یابد. ویژگی هایی مانند در آمد، سال های تحصیل و نرخ مرگ و میر بزرگسالان اهمیت زیادی داشته اند در حالی که به نظر می رسد ویژگی هایی مانند نرخ مرگ و میر نوزادان، تعداد موارد سرخک و یا جمعیت تاثیر زیادی در این پیش بینی نداشته اند. همچنین می توان دید ویژگی های درآمد و سال های تحصیل با سن امید به زندگی هم بستگی مثبت دارند و هر چه مقدار آن ها بیشتر باشد تاثیر آن در سن امید به زندگی بیشتر است اما ویژگی ای مانند نرخ مرگ و میر دارای correlation منفی با سن امید به زندگی است و هر چه این مقدار کمتر باشد سن امید به زندگی بالاتر است که از دید انسانها نیز منطقی به نظر می رسد.

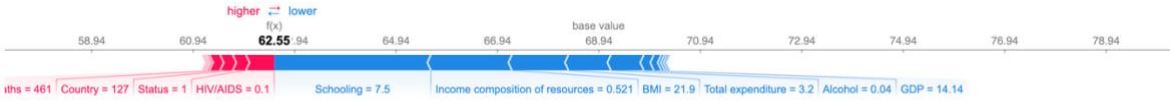
حال دو کشور ایران و پاکستان را در قاره ی آسیا به طور جداگانه بررسی کرده و نمودار force_plot را برای آنها رسم می کنیم.

مقدار سن امید به زندگی برای دو نمونه انتخاب شده به صورت زیر است:

paskistan: 65.1
iran: 72.0

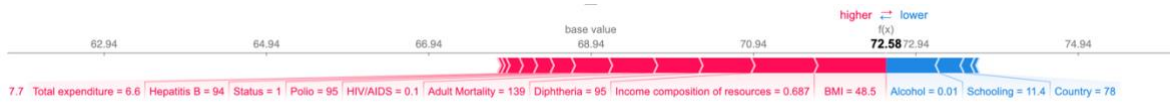
Kernel shap

pakistan:



شکل۴. نمودار force plot برای کشور پاکستان(kernel shap)

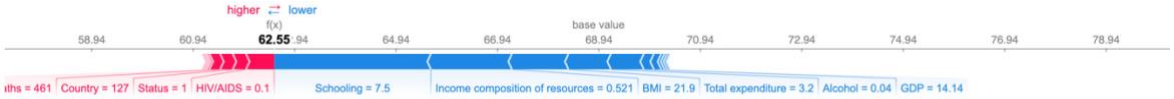
Iran:



شکل۵. نمودار force plot برای کشور ایران (kernel shap)

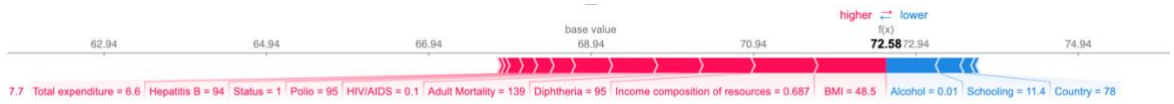
Deep shap

pakistan:



شکل۶. نمودار force plot برای کشور پاکستان(deep shap)

Iran:



شکل۷. نمودار force plot برای کشور ایران(deep shap)

مجددا مشاهده می شود که نمودار های ایجاد شده برای هر دو روش با یکدیگر هم خوانی دارند. برای کشور ایران می توان دید که ویژگی هایی مانند مصرف الکل، میزان تحصیلات و یا نام کشور باعث شده مقدار سن امید به زندگی به سمت پایین تر کشیده شود. در طرف مقابل ویژگی هایی مانند BMI، درآمد، ایمن سازی نسبت به دیفتری موجب بالا رفتن سن امید به زندگی نسبت به میانگین نمونه ها یا مقدار base value شده است.(هر چه طول روی محور بیشتر باشد نشان دهنده این است که ویژگی مورد نظر تاثیر مثبت یا منفی بیشتری داشته است)

برای کشور پاکستان نیز می توان گفت ویژگی هایی مانند میزان تحصیلات، میزان درآمد و BMI باعث پایین آمدن سن امید به زندگی نسبت به حالت میانگین شده است(می توان گفت احتمالا مقادیر این ویژگی ها برای کشور پاکستان به نسبت میانگین کشور ها کمتر بوده و همین موضوع باعث پایین آمدن سن امید به زندگی شده است). از طرفی ویژگی هایی مانند نرخ ایمن سازی HIV و در حال توسعه بودن این کشور باعث بالا رفتن سن امید به زندگی شده است. به طور کلی میتوان گفت در کشور پاکستان بیشتر ویژگی ها به سمت کاهش مقدار سن امید به زندگی بوده اند و به همین دلیل مقدار پیش بینی شده برای این کشور از میانگین نمونه ها کمتر شده است.

سوال 2 – Knowledge Distillation

در این سوال به توضیح یک روش knowledge distillation جهت تفسیر شبکه های عصبی پرداخته می شود.

۱. در این مقاله سعی شده با انتقال اطلاعات یک مدل شبکه عصبی مانند احتمالات در نظر گرفته شده برای هر ورودی به یک درخت تصمیم، رفتار آن شبکه عصبی را تقسیر کرد. در واقع در این جا شبکه عصبی به عنوان یک teacher برای درخت تصمیم عمل میکند. همان طور که میدانیم شبکه های عصبی می توانند در تسک ها به دقت بالایی برسند اما متاسفانه به دلیل وجود نورون ها و لایه های زیادی که بر هم ثاثیر میگذارند فهم رفتار این مدل ها بسیار دشوار است. از طرفی مدل هایی مانند درخت تصمیم تفسیر پذیری بالایی دارند اما نمی توانند به دقت خوبی برسند. بنابراین با آموزش توزیع های یادگرفته شده توسط شبکه عصبی به یک درخت تصمیم می توان به مدلی رسید که علاوه بر دقت نسبتا بالا توانایی تفسیر پذیری هم دارد. علاوه بر این استفاده از این روش به ما کمک می کند تا در صورتی که حجم زیادی داده برچسب نخورده داشته باشیم به کمک شبکه عصبی آن ها را برچسب بزنیم و به این ترتیب بتوانیم با داده بیشتری درخت تصمیم را آموزش دهیم. حتی می توان به کمک توزیع یادگرفته شده توسط شبکه داده های جدید تولید کرد و به این صورت اطلاعات زیادی را برای آموزش یک مدل درخت تصمیم به دست آورد.

۲. همان طور که در قسمت قبل نیز گفته شد این مدل دیگر معماری یک شبکه عصبی را ندارد که از یک سلسله از ویژگی ها برای تصمیم گیری استفاده کند، بلکه مدل پیشنهاد شده یک درخت تصمیم است که یاد گرفته است شبکه عصبی به ازای هر ورودی چه توزیع در خروجی تولید می کند یعنی در واقع سعی می کند تا تابع ورودی – خروجی شبکه عصبی را تقلید کند. بنابراین این مدل هنگام تصمیم گیری از یک سلسله از تصمیم ها حین مسیر مشخص شده توسط درخت استفاده می کند.

۳. مقدار loss به صورت زیر تعریف می شود:

$$\mathrm{L}(\mathbf{x}) = -\log\left(\sum_{\ell \in LeafNodes} P^{\ell}(\mathbf{x}) \sum_{k} T_k \log Q_k^{\ell}\right) \tag{9}$$

در فرمول بالا p^l احتمال رسیدن به نود l را مشخص می کند. Tهمان توزیع خروجی شبکه عصبی بر روی k کلاس است و Q^l توزیع خروجی هر نود بر روی k کلاس است. در صورتی که علامت منفی را به داخلی ترین سیگما ببریم داریم:

$$L(x) = \log\left(\sum_{l \in LeafNodes} P^{l}(x) \sum_{k} log\ \frac{Q_k^l}{T_k}\right) = \log\left(\sum_{l \in LeafNodes} P^{l}(x) \sum_{k} log\ (Q_k^l - T_K)\right) \tag{10}$$

عبارت بالا در واقع برای هر برگ محاسبه می کند که احتمال رسیدن به آن برگ با دیدن ورودی x چه قدر است و اگر در آن نود بودیم به ازای هر کلاس تفاوت مقدار مشخص شده توسط درخت را با مقدار احتمال نسبت داده شده توسط شبکه عصبی در نظر میگیرد و تلاش می کند که مجموع تمام این تفاوت ها را کمینه کند. یعنی در واقع در این جا سعی می کنیم که احتمالات نسبت داده شده به هر کلاس تا حد ممکن به احتمالات شبکه عصبی نزدیک باشد. این موضوع تقریبا شبیه به همان cross entropy loss است با این تفاوت که در cross entropy هدف این است که برچسب ها به برچسب واقعی نزدیک باشد اما در این جا هدف شبیه شدن توزیع اختصاص داده شده برای کلاس ها به توزیع خروجی شبکه عصبی است.

۴. علت اضافه کردن regularization برای جلوگیری از گیر کردن در مسیر های نادرست است. در واقع وجود این ترم باعث می شود تا در هر نود احتمال رفتن به زیر درخت سمت راست و چپ تقریبا برابر باشد. اگر این ترم وجود نداشت ممکن بود درخت در مسیر هایی گیر کند که در آن نودهای داخلی همواره بیشترین احتمال را به یکی از زیر درخت هایش بدهد

و در نتیجه در نهایت درخت به یک سمت بایاس پیدا می کند وگرادیان همواره نزدیک به صفر باقی می ماند که باعث عملکرد بد مدل می شود.

$$\alpha_i = \frac{\sum_{\mathbf{x}} P^i(\mathbf{x}) p_i(\mathbf{x})}{\sum_{\mathbf{x}} P^i(\mathbf{x})} \tag{11}$$

$$C = -\lambda \sum_{i \in InnerNodes} 0.5 \log(\alpha_i) + 0.5 \log(1 - \alpha_i) \tag{12}$$

در اینجا C همان ترم مربوط به regularization است. $P^i(x)$ احتمال رسیدن به نود i ام و $p_i(x)$ احتمال رفتن به شاخه سمت راست است. همان طور که مشاهده می شود با در نظر گرفتن ضریب 0.5 برای هر دو احتمال سعی می شود تا درخت از هر دو زیر درخت با احتمال مساوی مسیر را انتخاب کند.

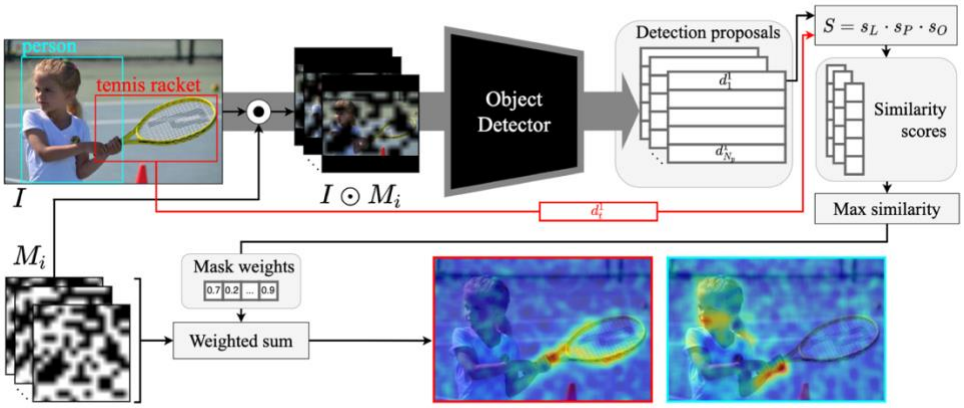
سوال 3 – D-RISE

در این سوال به بررسی روش D-RISE جهت تفسیر مدل ها در تسک object dectection پرداخته می شود.

(a

اخیرا متد های زیادی برای تفسیر مدل ها معرفی شده اند اگرچه تمرکز بیشتر این مدل ها بر روی تسک هایی مانند image classification است و توجه کمتری بر روی تسک های object detection است. علاوه بر این بیشتر این روش ها معمولا برای توضیح رفتار مدل نیاز به دانستن اطلاعاتی مانند معماری مدل مدل دارند که لزوما همیشه در دسترس نیست(روش های gradient base). همچنین در تسکی مانند object detector نه تنها احتیاج به توضیح مدل برای اختصای برچسب در هر محدوده داریم بلکه لازم است رفتار مدل برای انتخاب چنین محدوده ای نیز تحلیل کنیم که در مدل های پیشین ارائه شده خیلی به آن توجهی نمی شد(در object detection ممکن است احتیاج به توضیح چندین bounding box و class label داشته باشیم). متد D-RISE تلاش می کند تا به کمک روش های masking مدل های پیچده ای را در انجام تسک object detector تفسیر کند بدون آن که اطلاعاتی راجع به معماری مدل داشته باشد و یا از روش های مبتنی بر گرادیان کمک بگیرد. در واقع این روش به صورت black-box عمل می کند و می تواند برای هر نوعobject detector ای مورد استفاده قرار بگیرد.

تصویر زیر چگونگی عملکرد این مدل را نشان می دهد:



شکل۸. الگوریتم D-RISE

این روش به این صورت عمل می کند که به ازای هر تصویر تعدادی تصویر دیگر تولید می کند که هر کدام از این تصویر ها بخشی از تصویر اصلی را ندارند و یا با وضوح کمتری دارند. در ادامه هر یک از تصویر های جدید تولید شده به مدل داده می شود تا پیش بینی اش را بر مبنای ورودی برای تشخیص مکان اشیا و دسته بندی آن ها انجام دهد. در واقع خروجی تولید شره برای هر عکس شامل چندین وکتور است که هر کدام تعدادی محدوده و لیبل تولید می کنند. سپس از بین تمام پیش بینی های انجام شده پیش بینی انتخاب می شود که بیشترین شباهت را با پاسخ مطلوب ما داشته باشد و بسته به میزان شباهت یک ضریب برای ورودی مسک شده در نظر گرفته می شود. با انجام این کار برای تمام مسک های تولید شده و با استفاده از میانگین گیری وزن دار می توان مشخص کرد که کدام قسمت ها برای انجام پیش بینی مهم تر بوده اند.

(b

برای تولید مسک از مراحل زیر استفاده شده است:

۱. به تعداد N مسک به صورت ۰ و ۱ در یک پنجره به طول h*w تولید می شود(که کمتر از سایز تصویر وروی H*W است) و هر المان در این آرایه با احتمال مساوی مقدار ۰ یا ۱ را می پذیرد.

۲. سپس سایز هر یک از مسک های تولید شده را به کمک (bilinear)interpolation افزایش داده تا در انتها به یک پنجره به سایز $(h+1)CH \times (w+1)$ برسیم که در آن $CH \times CW = \lfloor H/h \rfloor \times \lfloor W/w \rfloor$

۳. برش تصویر به کمک مسک های تولید شده به طوری که آفست آن به صورت تصادفی از نقطه س (۰،۰) تا(CH,CW) انتخاب می شود.

(c

همان طور که پیش تر نیز گفته شد در تسک object detection علاوه بر اختصاص دادن برچسب به هر دسته لازم است تا مکان آن نیز مشخص شود. به این منظور ورودی های داده شده به مدل شامل سه بخش می باشند:

$$d_i = [L_i, O_i, P_i] = [(x^i_1, y1^i, x^i_2, y2^i), O_i, (p^i_1, \ldots, p^i_C)] \quad (13)$$

L_i اطلاعات مربوط به گوشه های در نظر گرفته شده برای مستطیل در نظر گرفته شده است.

O_i احتمال اینکه محدوده مشخص شده شامل هر نوع شی از هر نوع کلاسی باشد.

P_i یک وکتور از احتمالات که نشان میدهد احتمال اینکه محدوده در نظر گرفته عضو هر کلاس باشد چه قدر است.

معیار شباهت به کمک رابطه ی زیر به دست می آید:

$$s(d_t, d_j) = s_L(d_t, d_j) \cdot s_P(d_t, d_j) \cdot s_O(d_t, d_j), \tag{14}$$

$$\begin{aligned} s_L(d_t, d_j) &= \mathrm{IoU}(L_t, L_j), \\ s_P(d_t, d_j) &= \frac{P_t \cdot P_j}{\|P_t\| \|P_j\|}, \\ s_O(d_t, d_j) &= O_j. \end{aligned}$$

در واقع برای میزان شباهت محدوده مستطیلی از تقسیم مساحت مشترک به اشتراک مساحت ها استفاده می شود. برای مشابهت احتمالات از cosine similarity و برای وجود شی نیز از آن جا که میدانیم در تصویر هدف قطعا یک شی وجود دارد همان O_j را استفاده می کنیم. همچنین برای محاسبه شباهت کلی از حاصل ضرب اسکالر استفاده شده است که logical AND را مدلسازی می کند به این صورت که اگر مقادیر برای یکی از معیارهای شباهت پایین باشد شباهت کل نیز کم است.

(e

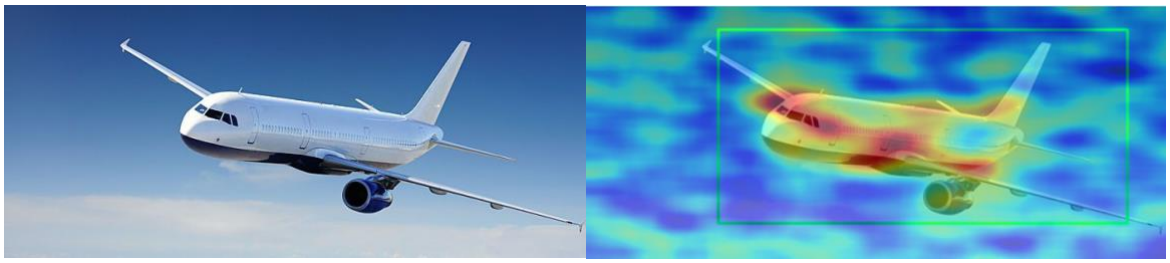
در ادامه به کمک notebook قرار داده شده روش زیر در سه تصویر تست شد که نتایج مربوط به هر کدام در زیر قابل مشاهده است.

شکل۹. اجرای D-RISE بر روی تصویر گربه

همان طور که مشخص است در اینجا مدل برای تشخیص گربه بیشتر به صورت توجه داشته است که کاملا منطقی است چرا که از نظر انسان ها نیز بیشترین اطلاعات راجع به گربه در صورت آن وجود دارد.

شکل ۱۰. اجرای D-RISE بر روی تصویر تابلوی ایست

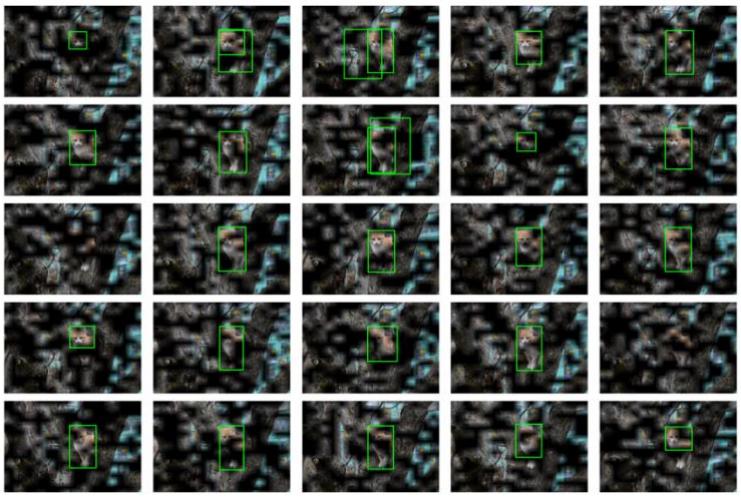
مجددا مشاهده می شود که روش پیشنهاد شده تقریبا به صورت منطقی عمل کرده است چرا که در تشخیص علامت ایست بیشتر توجه آن نقاطی بوده است که نوشته داشتند و باعث تفاوت می شوند و نه نقاط قرمز رنگ موجود در حاشیه مستطیل در نظر گرفته شده.



شکل ۱۱. اجرای D-RISE بر روی تصویر هواپیما

شاید بتوان گفت این تفسیر به خوبی تفسیر های قبلی موفق نبوده اما باز هم از آن جا که بیشتر فضای درون مستطیل آسمان هست و مدل به بخش با نسبت کوچکتر که همان هواپیماست توجه بیشتری داشته نشان می دهد باز هم به درستی و منطقی عمل کرده است.

تصویر زیر یک نمونه از تعدادی از مسک های تولید شده برای تصویر اول را نشان می دهد.



شکل ۱۲. نمونه ای از تصاویر مسک تولید شده برای تصویر گربه

سوال ۴ – LIME

در این سوال از روش LIME برای تفسیر یک مدل image classification استفاده می شود.

(a و b)

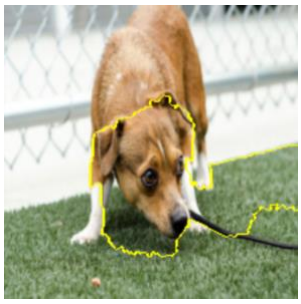
در این سوال از مدل آموزش داده شده MobileNet v2 استفاده می شود. برای تست مدل از یک تصویر سگ استفاده شده است که همان طور که می بینید خروجی های مدل نژاد های مختلف سگ را به عنوان ۵ خروجی با بیشترین احتمال باز می گرداند که نشان میدهد مدل به درستی کار میکند.

شکل۱۳. تصویر سگ

Labrador retriever 0.6585021018981934
American Staffordshire terrier 0.12135866284370422
golden retriever 0.05020306259393692
redbone 0.014067724347114563
dingo 0.012581673450767994

(c,d)

پس از تعریف مدل در نظر گرفته شده برای پکیج lime و به کمک پکیج skimage محدوده هایی که بیشتر به تصمیم گیری مدل کمک کرده است به دست می آید.

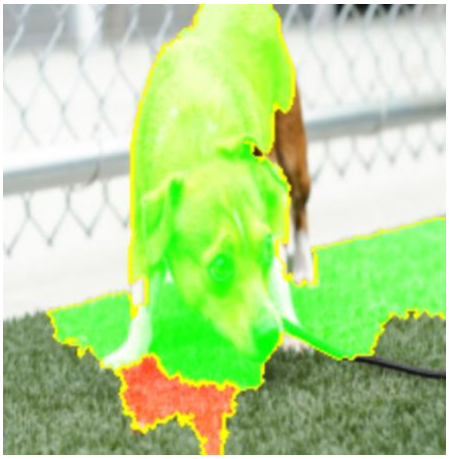


شکل۱۴. نواحی مشخص شده برای تصویر سگ توسط LIME

همان طور که انتظار میرفت مدل برای پیش بینی سگ از صورت آن استفاده کرده است. البته تکه ای از چمن هم مشاهده می شود که می تواند به این علت باشد که بیشتر عکس های سگ در محیط های آزاد و فضای سبز هستند و مدل نیز از این موضوع استفاده کرده است.

(e

در ادامه نواحی که به اطمینان بیشتری به مدل برای برچسب نهایی(Labrador retriever) داده است و تصمیم گیری را قوی تر کرده است قابل مشاهده است(نقاط سبز رنگ) همچنین می توان نقاطی که این تصمیم گیری را تضعیف کرده است نیز مشاهده کرد(نقاط قرمز رنگ).

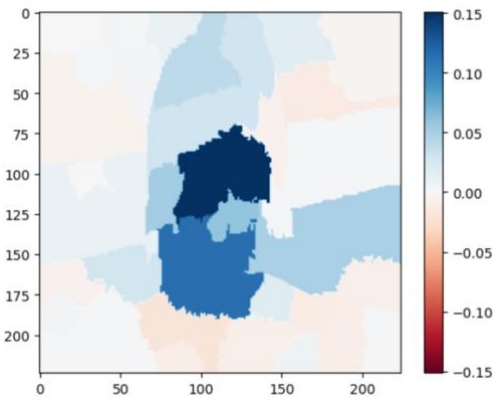


شکل۱۵. نواحی pros و cons برای برچسب سگ

میتوان گفت بیشتر نقاط سبز رنگ شامل سگ هستند و این نشان می دهد که مدل به خوبی محدوده سگ در تصویر را شناسایی کرده است. اگرچه مشاهده می شود که بر خلاف سبز بودن نقاطی از چمن، نقاط دیگر حاوی چمن در خلاف تصمیم گیری بوده اند.

(f

تصویر زیر به کمک یک heatmap اهمیت هر بخش از تصویر را برای پیش بینی با بیشترین احتمال نشان می دهد.



شکل۱۶. رسم heatmap برای تصویر سگ

در تصویر بالا هر از سمت قرمز به سمت آبی حرکت می کنیم اهمیت آن بخش تصویر افزایش می یابد. اکثر نقاطی که به رنگ آبی هستند مربوط به بدن و یا سر سگ هستند و سایر نقاط مانند فضای سفید رنگ و یا چمن تاثیر کمتری در تصمیم گیری مدل داشته اند.

(g,h

برای این بخش دو تصویر دیگر، یکی تصویر شامل گربه و دیگری شامل گربه و سگ نیز مورد بررسی قرار میگیرند و رفتار مدل توسط روش lime تفسیر می شود.

- **نتایج گربه**

شکل۱۷. تصویر گربه

Egyptian cat 0.5172398090362549
tabby 0.16973239183425903
tiger cat 0.11068528145551682
plastic bag 0.053881481289863586
lynx 0.04483482614159584

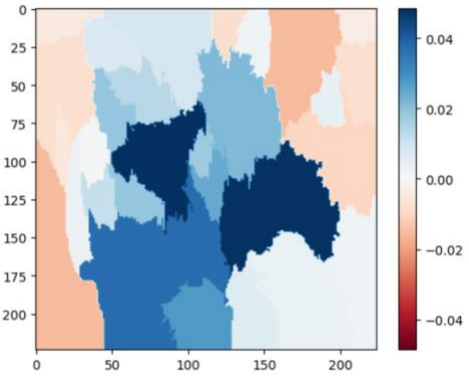
سه پیش بینی اول همه مربوط به نژاد های مختلف گربه هستند که عملکرد درست مدا را نشان میدهد.

شکل۱۸. نواحی مشخص شده برای تصویر گربه توسط LIME

بخش مهم برای انجام تصمیم گیری همانند حالت قبل بیشتر بر روی صورت تمرکز داشته است.

شکل۱۹. نواحی pros و cons برای برچسب گربه

نقاطی که در جهت تصمیم گیری برای کلاس گربه بودند محدوده های دربرگیرنده صورت گربه بودند و نقاط منفی نقاطی بوده اند که کاملا خارج از صورت و بدن گربه هستند که کاملا درست به نظر میرسد.



شکل ۲۰. رسم heatmap برای تصویر گربه

تصویر heatmap نیز به خوبی موارد ذکر شده در بخش های قبلی را نشان می دهد. همان طور که مشخص است، نواحی که در برگیرنده گربه بودند چه صورت و چه بدن به رنگ آبی هستند که نشان میدهد مدل بیشتر برای تصمیم گیری از آن ها استفاده کرده است. در حالی که سایر نواحی خارج از گربه به رنگ قرمز هستند به این معنی که تاثیرات منفی و یا خیلی کمی بر روی تصمیم گیری داشته اند.

- **نتایج گربه و سگ**

شکل۲۱. تصویر گربه و سگ

beagle 0.42868441343307495
carton 0.19388310611248016
Brittany spaniel 0.02289739064872265
Egyptian cat 0.01852933131158352
bow tie 0.01694798283278942

در تصویر بالا پیش بینی اول مدل با بیشترین اطمینان سگ بوده است اگرچه به دلیل وجود گربه در تصویر پیش بینی چهارم مربوط به یک گربه است که نشان می دهد مدل تا حدی متوجه وجود گربه در تصویر هم شده است.

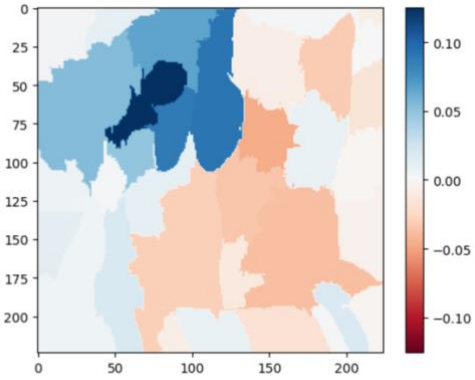
شکل۲۲. نواحی مشخص شده برای تصویر گربه و سگ توسط LIME

نواحی تاثیر گذار در انتخاب کلاس سگ مجددا صورت سگ بوده است و می بینیم که نواحی شامل گربه در این تصمیم گیری موثر نبوده اند و مدل به آن ها توجهی نداشته است.

شکل۲۳. نواحی pros و cons برای برچسب سگ درتصویر گربه و سگ

با رسم نقاط prosوcons بر روی تصویر به وضوح این موضوع دیده می شود که نقاطی که شامل گربه بوده اند مدل را دچار سختی در تصمیم گیری کرده اند چرا که بین سگ و گربه که در تصویر موجود بوده مدل دچار ابهام برای تصمیم گیری می شود و وجود گربه به ضرر تصمیم نهایی مدل بوده است در حالی که تصویر صورت سگ در جهت تصویر نهایی بوده است. در صورتی که این این نقاط pros و cons را برای تصمیم گربه رسم کنیم می بینم که کاملا رنگ نقاط به نفع گربه عوض می شود و محدوده سگ تضعیف کننده تصمیم هستند.

شکل۲۴. نواحی pros و cons برای برچسب گربه درتصویر گربه و سگ



شکل۲۵. رسم heatmap برای تصویر گربه و سگ

مجددا تصویر بالا نشان می دهد هنگام اختصاص برچسب سگ به تصویر، بیشتر توجه مدل بر روی سر سگ بوده است و نقاط تضعیف کننده شامل نواحل حاوی گربه بوده اند.