

# تشخیص تومور مغزی

## با استفاده از Transfer Learning

### بررسی کاربرد گرایانه

امیرحسین سلطانیپور، نیکو پاکنیا، حسین جعفری

یادگیری ماشین

دانشگاه شهید بهشتی

#### خلاصه

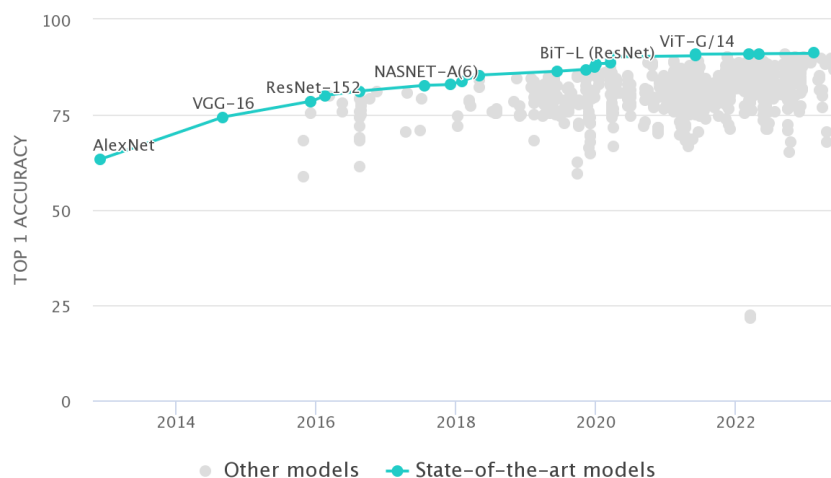
از وقتی ImageNet ساخته شد، مدل های بزرگی آمدند و در چالش image classification قدرت نمایی کردند. ما با یکی از مسائل کلاسیک این حوزه، ابتدا به بررسی این مدل ها می پردازیم. بهینه ترین را برگزیده و به پیاده سازی آن به صورت یک وب اپلیکیشن دست می بریم. دو راه اصلی برای ساختار همچین اپلیکیشنی را بررسی کرده سپس یک راه بهتر سومی را می سازیم. در آخر به عنوان کار های آینده، راه حلی برای ارائه دقیق تر خروجی شبکه عصبی به پزشک بیان می کنیم.

## ۱. ImageNet

ImageNet یک دیتاست شامل ۱۴,۱۹۷,۱۲۲ عکس لیبل زده شده است. از سال ۲۰۱۰ این دیتاست به عنوان benchmark در حوزه image classification و object detection استفاده می‌شود. سالانه چالشی با این دیتاست برگزار می‌شود تحت عنوان ILSVRC<sup>۱</sup>.

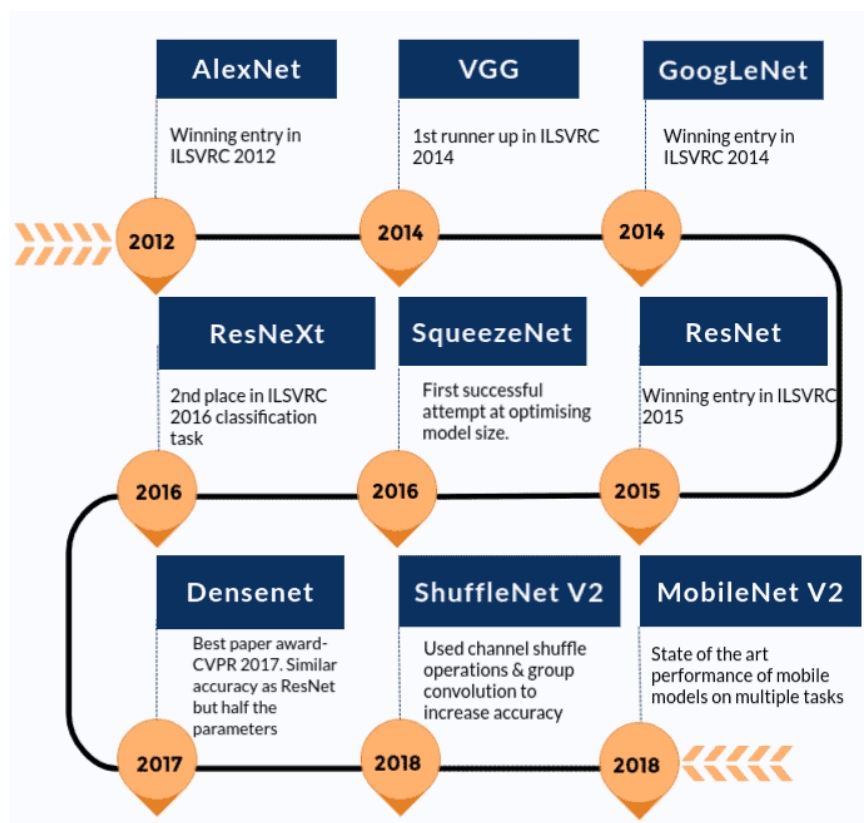
## ۲. مدل های مشهور

مدل های train شده متفاوتی روی ImageNet برای مسئله image classification وجود دارد. این مدل ها با معیار های مختلفی سنجیده می‌شوند. اولین معیار accuracy است. در نمودار زیر برترین مدل سال را بر این اساس می‌بینید.



<sup>1</sup> ImageNet Large Scale Visual Recognition Challenge

از معیار های دیگر می توان به performance اشاره کرد. یا حجم مدل. در شکل زیر مدل های مشهوری را به ترتیب سال مشاهده می کنید.



### ۳. مدل های انتخابی ما

ما سه مدل را برگزیدیم برای بررسی و اجرا کردن.

- VGG-16
- ResNet
- MobileNet

این سه از پر استفاده ترین مدل های pre trained است. VGG-16 و ResNet را در نمودار بالا می بینید. این دو در زمانی که ارائه شدند، نوع آوری های شگرفی به این عرصه هدیه دادند. با گسترش بیشتر کاربرد این مدل ها، به مدل های بهینه تر و سریع تر نیاز پیدا کردیم که روی دیوایس ها در عمل کاربردی باشند. به این منظور مدل MobileNet یکی از معروف ترین ها می باشد.

#### ۴. VGG-16

این مدل<sup>۲</sup> توسط Zisserman و Simonyan ارائه شد. از گروه Visual Geometry Group (VGG). از دانشگاه آکسفورد. این مدل به kernel ساده آن مشهور است. convolutional layer های ۳ در ۳. هر چه بیشتر در عمق شبکه فرو می رویم، شبکه سنگین تر می شود. در زیر معماری آن را می بینید.

---

<sup>2</sup> *Very deep convolutional networks for large scale image recognition*

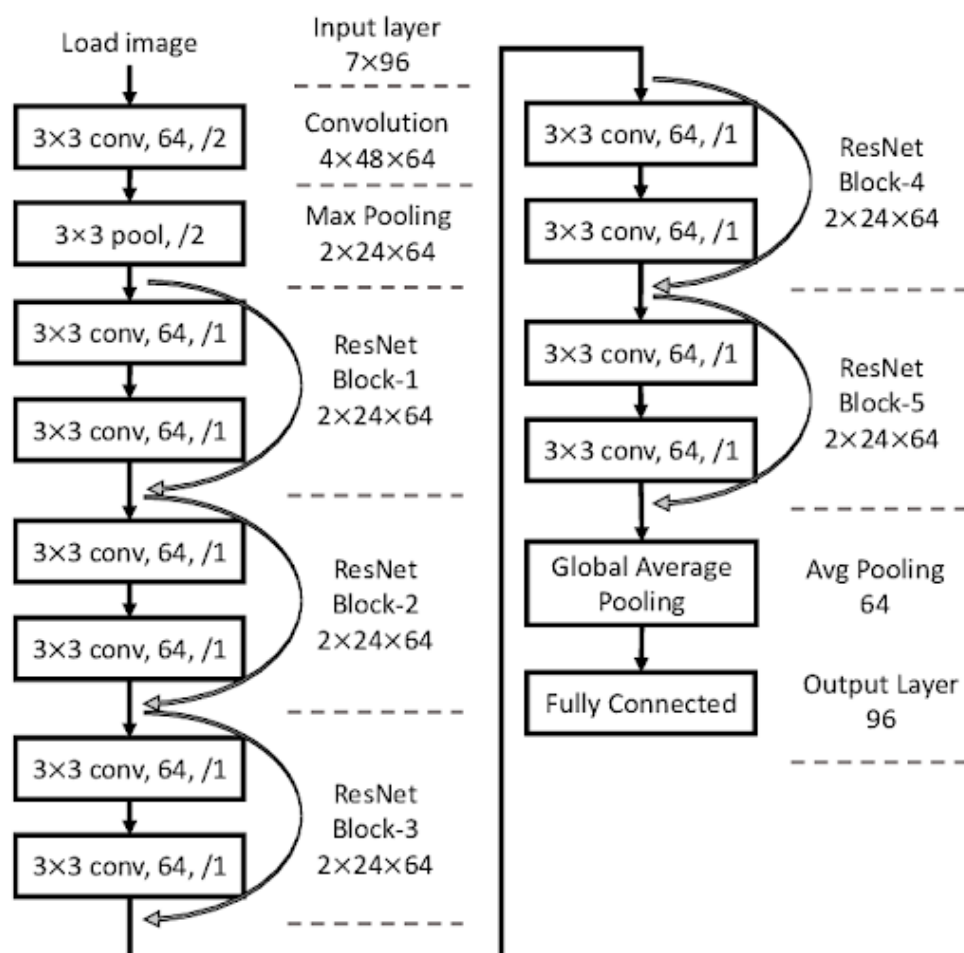
ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input ( $224 \times 224$ RGB image)					
conv3-64	conv3-64 <b>LRN</b>	conv3-64 <b>conv3-64</b>	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 <b>conv3-128</b>	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 <b>conv1-256</b>	conv3-256 conv3-256 <b>conv3-256</b>	conv3-256 conv3-256 conv3-256 <b>conv3-256</b>
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 <b>conv1-512</b>	conv3-512 conv3-512 <b>conv3-512</b>	conv3-512 conv3-512 conv3-512 <b>conv3-512</b>
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 <b>conv1-512</b>	conv3-512 conv3-512 <b>conv3-512</b>	conv3-512 conv3-512 conv3-512 <b>conv3-512</b>
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Table 2: Number of parameters (in millions).

Network	A,A-LRN	B	C	D	E
Number of parameters	133	133	134	138	144

## ۵. ResNet

این مدل توسط Kaiming He در سال ۲۰۱۵ ارائه شد. این مدل شبکه پیچیده‌ای دارد و sequential نیست. شبکه از به هم پیستن بلاک‌های کوچکتری تشکیل می‌شود. با این که تعداد لایه‌های بیشتری نسبت به VGG دارد، تعداد وزن‌های کمتری دارد. در شکل زیر معماری آن را می‌بینید.



## ۶. MobileNet

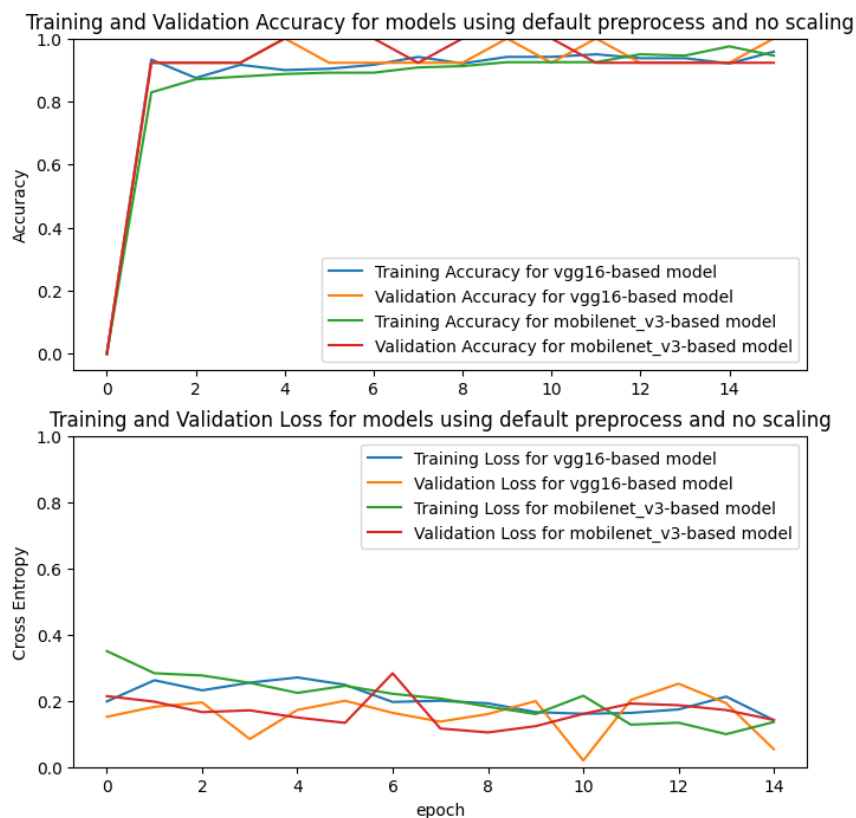
این مدل با accuracy مدل‌های قبلی، performance بهتری ارائه می‌دهد. برای سبک کردن شبکه از مفهومی به نام depth-wise separable convolution استفاده شده است. در شکل زیر معماری این مدل را می‌بینیم.

Table 1. MobileNet Body Architecture

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32 \text{ dw}$	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64 \text{ dw}$	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128 \text{ dw}$	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128 \text{ dw}$	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256 \text{ dw}$	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256 \text{ dw}$	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
$5 \times$	Conv dw / s1 $3 \times 3 \times 512 \text{ dw}$	$14 \times 14 \times 512$
	Conv / s1 $1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512 \text{ dw}$	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024 \text{ dw}$	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool $7 \times 7$	$7 \times 7 \times 1024$
FC / s1	$1024 \times 1000$	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

## ۷. این سه مدل در پروژه ما

با استفاده از این مدل ها به عنوان base و اتصال چندین لایه به انتهای آن ها (Transfer Learning) شبکه ای برای تشخیص تومور مغزی ساخته ایم. برای تفصیل تفاوت این سه به notebook پروژه رجوع شود. برای مثال تفاوت Loss و Accuracy دو مدل VGG-16 و MobileNet را در این پروژه را در زیر می بینیم.





## ۸. وب اپلیکیشن

وب اپلیکیشنی طراحی شده است که تصویر MRI مغز را می‌گیرد و در خروجی می‌گوید تومور وجود دارد یا خیر. مشابه های این سناریو خیلی استفاده می‌شود. تصویری از کاربر گرفته می‌شود و مدلی قراراست روی آن استنتاج کند. سه معماری برای همچین سناریوهایی وجود دارد.

۱. تصویر از دیوایس کاربر گرفته شود و اینترنتی به سروری منتقل شود. در سرور این تصویر پردازش شود. یعنی استنتاج شود که تومور هست یا نه. در نهایت نتیجه به دیوایس کاربر برگردانده شود و در آنجا نمایش داده شود.

۲. تصویر در همان دیوایس کاربر پردازش شود و نتیجه local به دست آورده شود. خوشبختانه با Tensorflow js ما توانایی همچین کاری داریم. زیرا زبان قابل اجرای browser ها، js<sup>۳</sup> است.

یک مشکل راه حل اول محدودیت پهنای باند است. مشکل دیگر راه اول محدودیت پردازشی سرور است که با افزایش کاربران چشمگیر می‌شود. اما راه حل دوم مشکلات دیگری دارد. به طور مثال محرمانگی مدل از بین می‌رود. مثلاً مدل تشخیص قلب

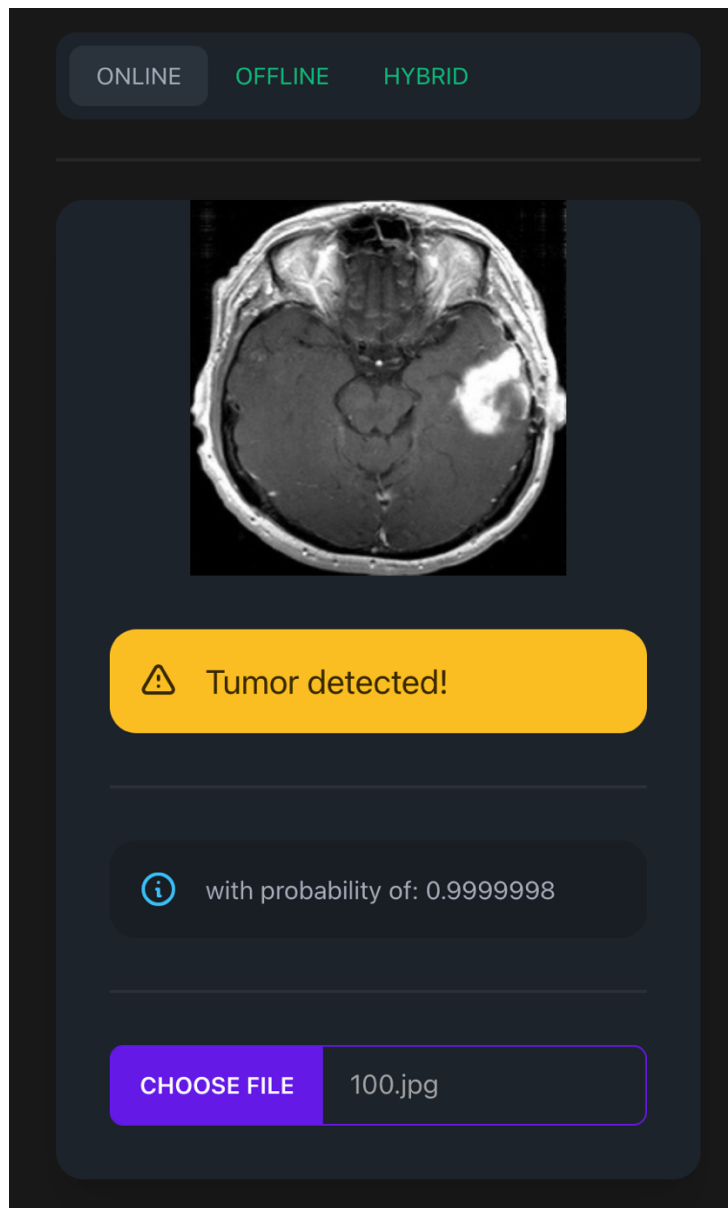
---

<sup>۳</sup> Javascript

۳. ما یک راه حل سومی ارائه داده‌ایم. ایده این است که بخشی از استنتاج در دیوایس کاربر و بقیه در سرور انجام شود. به این صورت:

۴. در اول شبکه یک Base Model داریم مثل VGG. ابتدا تصویر را در دیوایس کاربر به صورت local به این Base Model دهیم و نتیجه را دریافت کنیم. البته لایه classification آخر را برداشته ایم و نتیجه ما یک image feature vector است. از طرفی یک مدل کامل train شده در سرور است که شامل همین مدل Base Model و لایه های اضافه شده است. این مدل کاملاً به تنهایی کار می‌کند ولی برای مرحله استنتاج و بعد از مرحله train کردن، ما مدل Base را در سرور برمی‌داریم. حالا ما در سرور صرفاً لایه های train شده اضافه شده را داریم. نتیجه گرفته شده از دیوایس کاربر را به این لایه ها می‌دهیم و نتیجه نهایی در سرور به دست می‌آید و به دیوایس کاربر برگردانده می‌شود. به دلیل اینکه image feature vector خیلی سبک تر از خود عکس است، پهنای باند مورد نیاز به شدت کاهش می‌یابد. هم سرور هم کاربر. محرمانگی هم باقی می‌ماند.

ما هر سه راه را پیاده کردیم. تصویری از اپلیکیشن را  
در زیر می بینید.



در بالا این عکس سه دکمه می بینید. که به ترتیب از چپ به راست مربوط می شود به سه راه توضیح داده شده.

کد یک سرور هم نوشته شده که در ارتباط با این اپلیکیشن است.

قسمت فرانت با یک فریم ورک مشهور فرانت به اسم vue نوشته شده است. به این دلیل که از component های reusable ساخته شود تا مردم بتوانند از قست های این پروژه استفاده کنند.

نشانی گیت هاب این اپلیکیشن به صورت زیر است.

<https://github.com/AmirHossein-STP/vue-net/>

## ۹. کار آینده, Confidence Calibration

وقتی نتیجه تخمین یک عکس را به دکتر می گوئیم، خوب است که صرفاً آری یا خیر نباشد. این گونه که احتمال درستی این نتیجه گیری هم به دکتر داده شود. برای این کار می شود دقت و یا هر معیار ارزشیابی مدل به دکتر داده شود. راه بهتر این است که برای هر عکس احتمال درستی استنتاج مدل داده شود. مثلاً بگوئیم به احتمال ۹۹ درصد تومور وجود دارد. خب این احتمال را از روی خروجی مدل میتوانیم به دست بیاوریم. ولی اگر دقیق تر به عددی که مدل به ما میگوید فکر کنیم متوجه مشکلی می شویم. وقتی به دکتر گفته میشود ۹۹ درصد، او چه برداشتی می کند؟ انتظار دارد

که در بلند مدت به طور تقریبی ۹۹ درصد کسانی که احتمال تومور داشتندشان ۹۹ بوده، واقعا تومور داشته باشند. ولی آیا عددی که مدل به ما می‌دهد این ویژگی را دارد؟ خیر. به طور مثال مدل خیلی وقت‌ها می‌گوید ۹۹.۹۹ درصد احتمال درستی. مدل خوبی هم هست. با accuracy ۹۹ درصد مثلاً. فعلاً برای سادگی فرض میکنم تعداد اعضای کلاس‌ها تقریباً هم اندازه‌اند. که بگوییم دقت خوبی است. میدانیم حدوداً در بلند مدت از ۱۰۰ تا ۱ خطا دارد. ولی می‌گوید احتمال درستی ۹۹.۹۹ درصد. راه‌هایی وجود دارد برای تراز کردن احتمال گفته شده شبکه با احتمال تکرار شونده در دنیای واقعی. به این کار می‌گویند Confidence Calibration. در این پروژه فرصت نشد به این موضوع پرداخته شود ولی به نظر ما ویژگی مهمی است برای کاربردی شدن تشخیص‌های پزشکی. لینک از مقاله‌ای معروف در این حوزه در زیر آورده شده است.

On calibration of modern neural networks

<http://proceedings.mlr.press/v70/guo17a/guo17a.pdf>