

Explainable-AI - Post-Hoc-Analyse - Erkennung von Hirntumoren

Sven Sendke

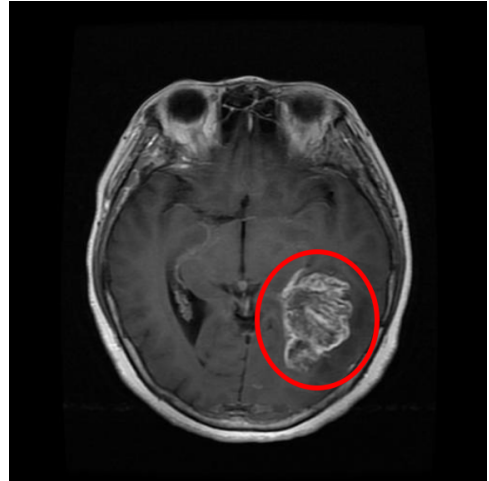
09.12.2024

Duale Hochschule Baden-Württemberg Stuttgart

1. Gliederung
2. Einleitung
3. Projekt
4. LRP
5. Beobachtungen
6. Fazit

Einleitung

- **Abnormales** Wachstum von **Zellen** im Gehirn
- In Deutschland erkranken **jährlich** etwa **8.000 Menschen** neu an einem Hirntumor
- **Zeitaufwändige** und **teure** Untersuchungsmethode



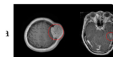
- Einsatz von CNNs in der **medizinischen Bildsegmentierung**
- Erkennung von Hirntumoren in MRT-Bildern ist schwierig aufgrund der **komplexen Gehirnstruktur**
- Schwierigkeit, die **Gründe** für die Antwort von CNN zu verstehen

Erstellung eines CNNs zur **Erkennung von Hirntumoren** und Anwendung von **Layer-wise Relevance Propagation** (LRP) zur Analyse der Relevanz der einzelnen Pixel für die **Entscheidungsfindung** des Netzwerks, um herauszufinden, welche **Muster** bei der Erkennung geholfen haben.

2.4 Einleitung - Datensatz

Brian Tumor Dataset

X-Ray images of Brain



| image | class |
|-----------------------|------------|
| filename | label |
| 4600 unique values | tumor 55% |
| | normal 45% |

<https://www.kaggle.com/datasets/preetviradiya/brian-tumor-dataset/data>

Probleme mit dem Datensatz:


- Doppelte Bilder
- Einige Bilder sind kleine Variationen



Achtung!




Auf der Seite steht, es seien Röntgenbilder, aber es sind MRT / CT Bilder!!!!


Projekt



3.1 Projekt - GitHub Repository








 **XAI-BrainTumor** Public

 Pin  Unwatch 1

 main  1 Branch  0 Tags



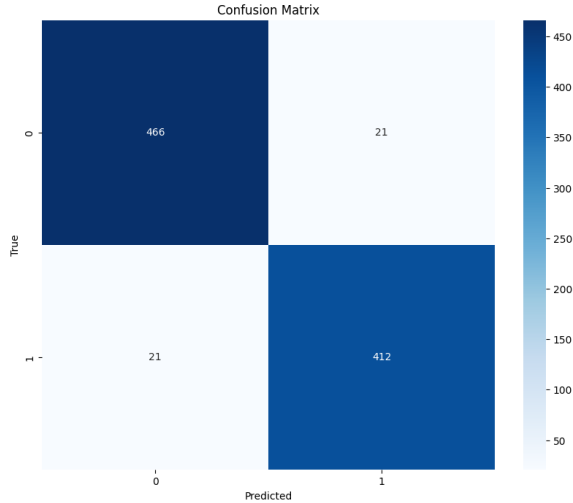
 Add file  Code

| | | |
|--|----------------------------------|--|
|  einBuegelbrett refactor: rename trained_net.pth | e8ea7cb · yesterday |  19 Commits |
|  images-documentation | docs: finish lrp documentation | yesterday |
|  lrp | docs: finish lrp documentation | yesterday |
|  .gitignore | feat: create CNN with PyTorch | 2 months ago |
|  README.md | refactor: rename trained_net.pth | yesterday |
|  main.ipynb | refactor: rename trained_net.pth | yesterday |

```
class Net(nn.Module):
    def __init__(self):
        super(Net, self).__init__()
        self.conv1 = nn.Conv2d(1, 10, kernel_size=5)
        self.relu1 = nn.ReLU()
        self.conv2 = nn.Conv2d(10, 20, kernel_size=5)
        self.relu2 = nn.ReLU()
        self.flatten = nn.Flatten()
        self.fc1 = nn.Linear(20 * 248 * 248, 50)
        self.relu3 = nn.ReLU()
        self.fc2 = nn.Linear(50, 10)
        self.fc3 = nn.Linear(10, 2) # 2 Classes (Brain Tumor / Healthy)

    def forward(self, x):
        x = self.conv1(x)
        x = self.relu1(x)
        x = self.conv2(x)
        x = self.relu2(x)
        x = self.flatten(x)
        x = self.fc1(x)
        x = self.relu3(x)
        x = self.fc2(x)
        x = self.fc3(x)
        return x
```

3.3 Projekt - Konfusionsmatrix



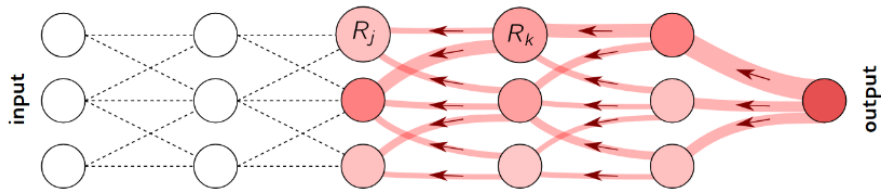
LRP

Layer-wise Relevance Propagation (LRP):

- Relevanz von jedem Neuron in ein Neuronales Netz
- Forward pass (Vorhersage)
- Output wird Schicht für Schicht rückwärts propagiert (Spezielle LRP Regeln)
- Heatmap mit dem Beitrag der einzelnen Input-Features (Pixel) zur Vorhersage

Hier noch einmal als Bild:

$$R_j = \sum_k \frac{z_{jk}}{\sum_j z_{jk}} R_k.$$



```
...  
├── lrp  
│   ├── __init__.py  
│   ├── lrp.py  
│   ├── lrp_filter.py  
│   └── lrp_layers.py  
└── ...
```



```
def forward(self, x: torch.tensor) -> torch.tensor:
    activations = list()

    with torch.no_grad():
        activations.append(torch.ones_like(x))

        for layer in self.layers:
            x = layer.forward(x)
            activations.append(x)

    activations = activations[::-1]
    activations = [a.data.requires_grad_(True) for a in activations]
    relevance = torch.softmax(activations.pop(0), dim=-1)

    for i, layer in enumerate(self.lrp_layers):
        relevance = layer.forward(activations.pop(0), relevance)

    return relevance.permute(0, 2, 3, 1).sum(dim=-1).squeeze().detach().cpu()
```

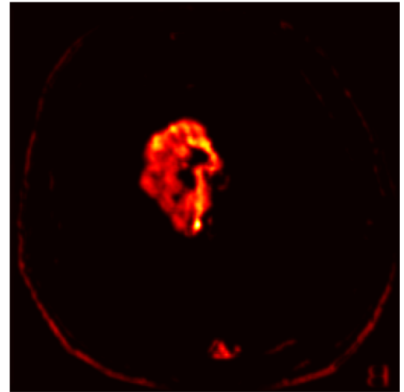
Beobachtungen

Gehirntumor
erkannt

Original Image
Predicted: Brain Tumor

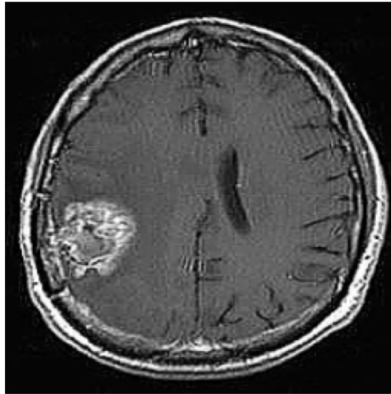


Relevance Map
Predicted: Brain Tumor

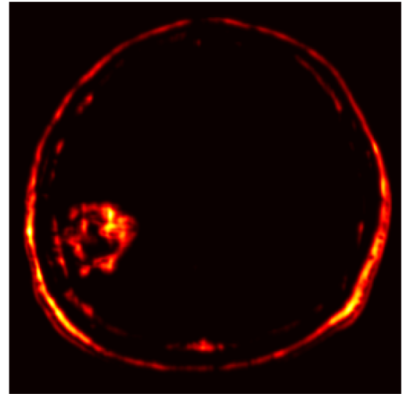


Gehirntumor
erkannt (mit
Hirnhaut)

Original Image
Predicted: Brain Tumor

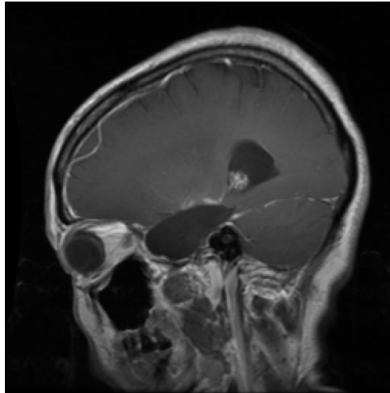


Relevance Map
Predicted: Brain Tumor



Auge/Tumor?
→ wenige
Datensätze
mit Augen

Original Image
Predicted: Brain Tumor



Relevance Map
Predicted: Brain Tumor

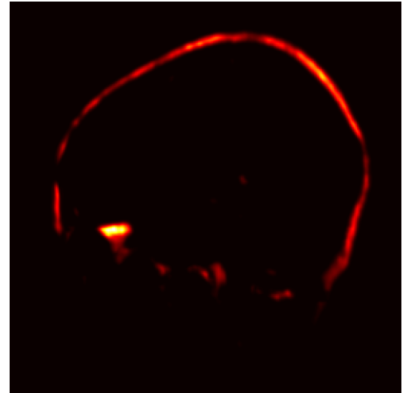
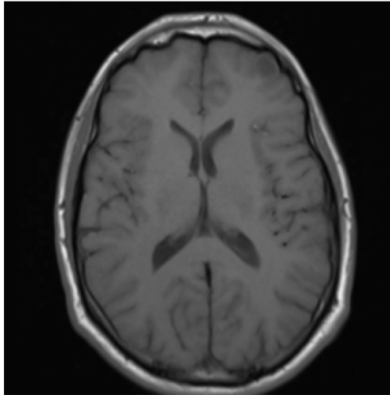


Bild mit
wenigen
Details
→ Hirnhaut,
ohne Auffäl-
ligkeiten

Original Image
Predicted: Healthy



Relevance Map
Predicted: Healthy

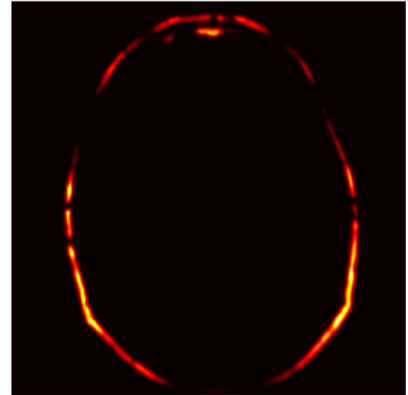
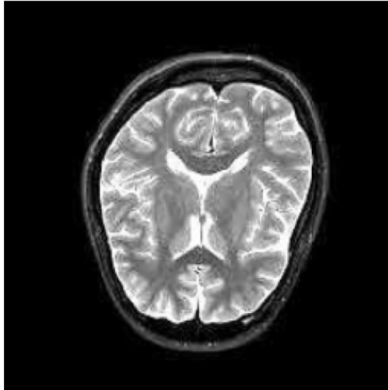
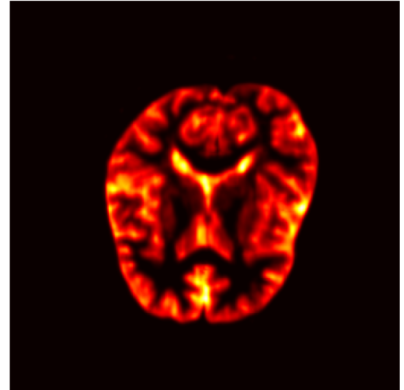


Bild mit
vielen Details
→
Gehirnform
entscheidend

Original Image
Predicted: Healthy



Relevance Map
Predicted: Healthy



Fazit

Vorteile:

- Relevante Merkmale als Heatmap
- Die Gesamtrelevanz auf allen Ebenen bleibt erhalten

Nachteile

- Ähnliche Merkmale können unabhängig von der untersuchten Klasse hervorgehoben werden
- Irrelevante Informationen können hervorgehoben werden

Erfolge:

- Klare Heatmaps
- "Mathe" umsetzen

Schwierigkeiten:

- Datensatz
- Optimierung CNN: zuerst 70% dann 95%

- [1] Image Processing Techniques for Brain Tumor Detection: A Review
- [2] CNN-based Segmentation of Medical Imaging Data
- [3] On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation
- [4] Layer-Wise Relevance Propagation for Deep Neural Network Architectures
- [5] Magnetic Resonance Imaging (MRI)
- [6] Layer-wise Relevance Propagation
- [7] Brain MRI
- [8] Towards Best Practice in Explaining Neural Network Decisions with LRP
- [9] Layer-Wise Relevance Propagation: An Overview
- [10] Interpreting the Predictions of Complex ML Models by Layer-wise Relevance Propagation
- [11] Explaining Convolutional Neural Networks using Softmax Gradient Layer-wise Relevance Propagation
- [12] Brain Cancer Screening

Vielen Dank für ihre Aufmerksamkeit!