By AI

Intracranial Hemorrhage Detection

Yuming Chen

Bo Feng

兰州大学 LANZHOU UNIVERSITY

Drexel UNIVERSITY

Project Overview

Overview

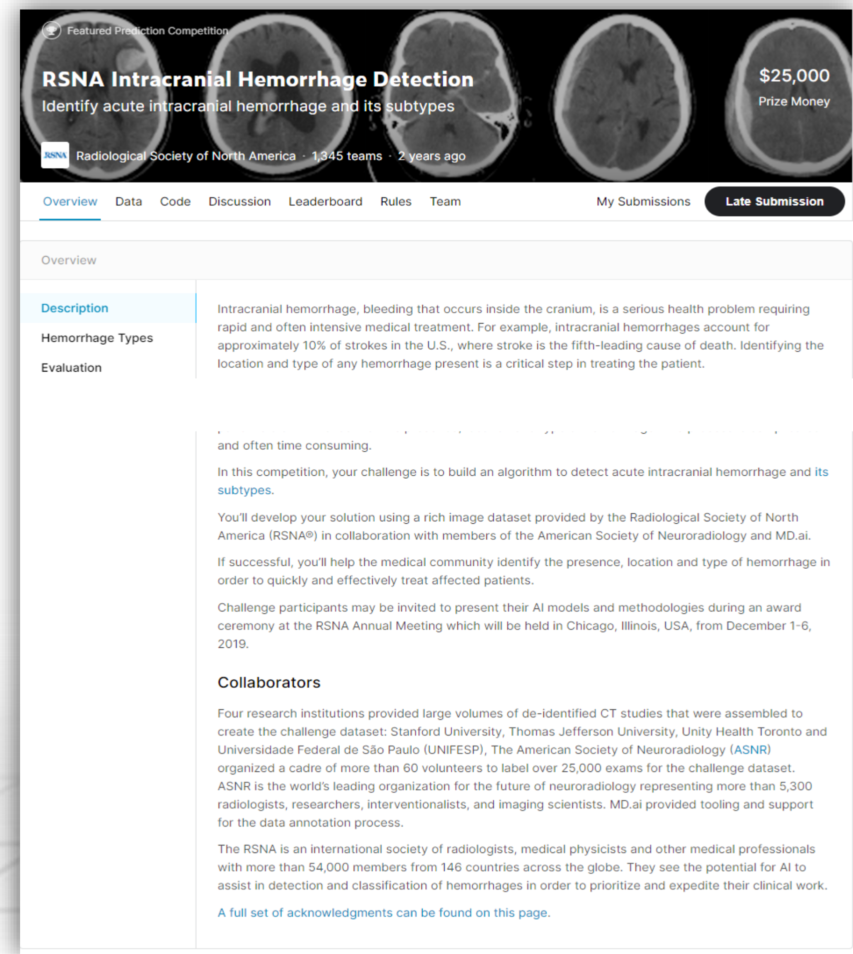
- **Serious Health Problem**
- **Rapid & Intensive Treatment**
- **Complicated Process**

⬇

- **Detection Subtypes(5)**
- **Multi–class Classification**
- **InceptionNetV3 & Spark**

Data Information

Data Origin



Featured Prediction Competition

RSNA Intracranial Hemorrhage Detection

Identify acute intracranial hemorrhage and its subtypes

$25,000
Prize Money

Radiological Society of North America · 1,345 teams · 2 years ago

Overview | Data | Code | Discussion | Leaderboard | Rules | Team | My Submissions | Late Submission

Overview

- Description
- Hemorrhage Types
- Evaluation

Intracranial hemorrhage, bleeding that occurs inside the cranium, is a serious health problem requiring rapid and often intensive medical treatment. For example, intracranial hemorrhages account for approximately 10% of strokes in the U.S., where stroke is the fifth-leading cause of death. Identifying the location and type of any hemorrhage present is a critical step in treating the patient.

and often time consuming.

In this competition, your challenge is to build an algorithm to detect acute intracranial hemorrhage and its subtypes.

You'll develop your solution using a rich image dataset provided by the Radiological Society of North America (RSNA®) in collaboration with members of the American Society of Neuroradiology and MD.ai.

If successful, you'll help the medical community identify the presence, location and type of hemorrhage in order to quickly and effectively treat affected patients.

Challenge participants may be invited to present their AI models and methodologies during an award ceremony at the RSNA Annual Meeting which will be held in Chicago, Illinois, USA, from December 1-6, 2019.

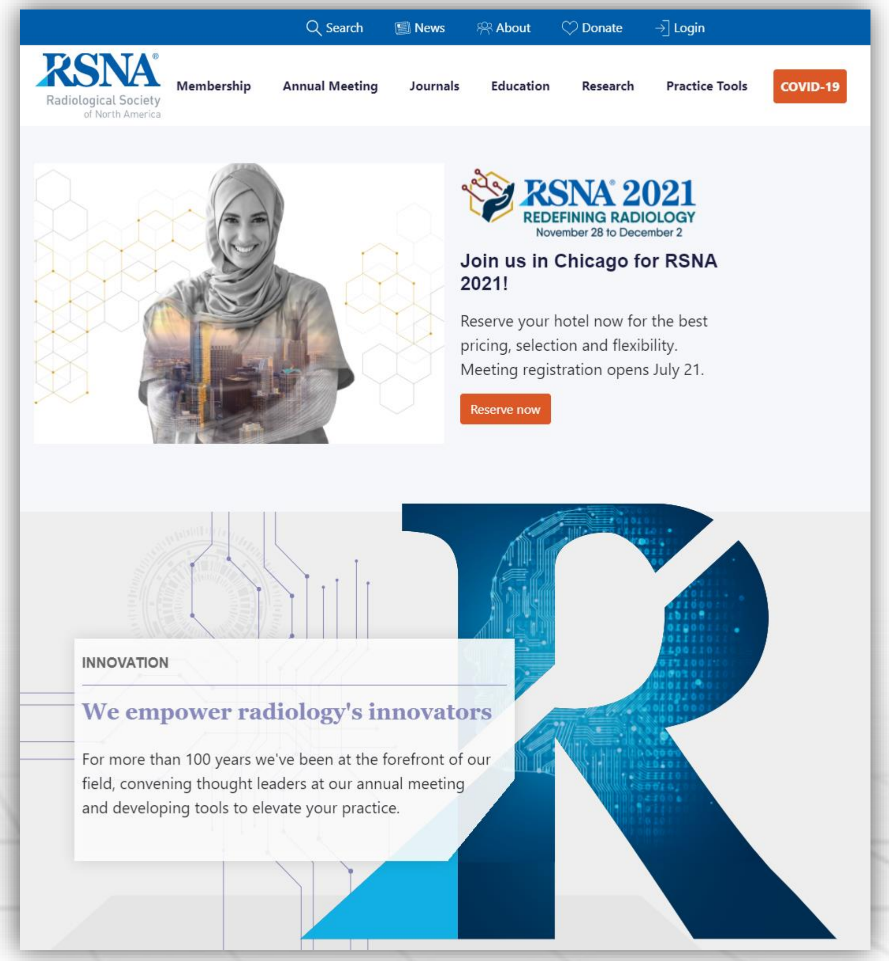
Collaborators

Four research institutions provided large volumes of de-identified CT studies that were assembled to create the challenge dataset: Stanford University, Thomas Jefferson University, Unity Health Toronto and Universidade Federal de São Paulo (UNIFESP), The American Society of Neuroradiology (ASNR) organized a cadre of more than 60 volunteers to label over 25,000 exams for the challenge dataset. ASNR is the world's leading organization for the future of neuroradiology representing more than 5,300 radiologists, researchers, interventionalists, and imaging scientists. MD.ai provided tooling and support for the data annotation process.

The RSNA is an international society of radiologists, medical physicists and other medical professionals with more than 54,000 members from 146 countries across the globe. They see the potential for AI to assist in detection and classification of hemorrhages in order to prioritize and expedite their clinical work.

A full set of acknowledgments can be found on this page.

kaggle **www.kaggle.com**



Search | News | About | Donate | Login

RSNA Radiological Society of North America

Membership | Annual Meeting | Journals | Education | Research | Practice Tools | COVID-19

RSNA 2021
REDEFINING RADIOLOGY
November 28 to December 2

Join us in Chicago for RSNA 2021!

Reserve your hotel now for the best pricing, selection and flexibility. Meeting registration opens July 21.

Reserve now

INNOVATION

We empower radiology's innovators

For more than 100 years we've been at the forefront of our field, convening thought leaders at our annual meeting and developing tools to elevate your practice.

RSNA® **www.rsna.org**

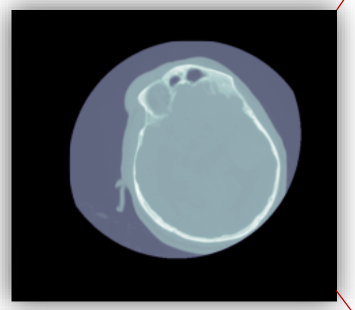
Data Format

CSV

19877

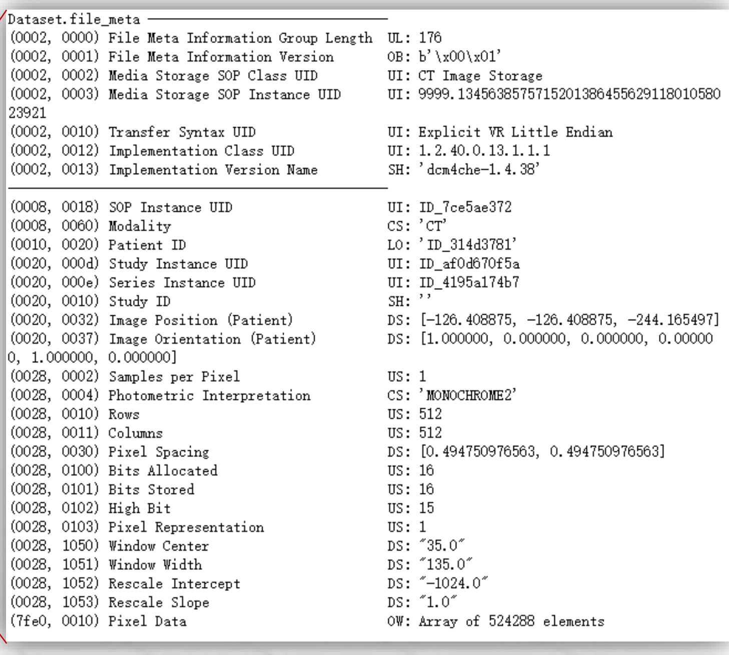
id	label
ID_52c205043	epidural
ID_15e6dc40a	intraventricular
ID_01a97add4	intraparenchymal
ID_9fed7e3db	subdural
ID_903d2ef48	subdural
ID_e0942808d	epidural
ID_d4d31f695	subdural
......	
ID_c64f8db24	subarachnoid
ID_d202babf2	subarachnoid
ID_9ffd25faa	subdural

DICOM



0.53MB

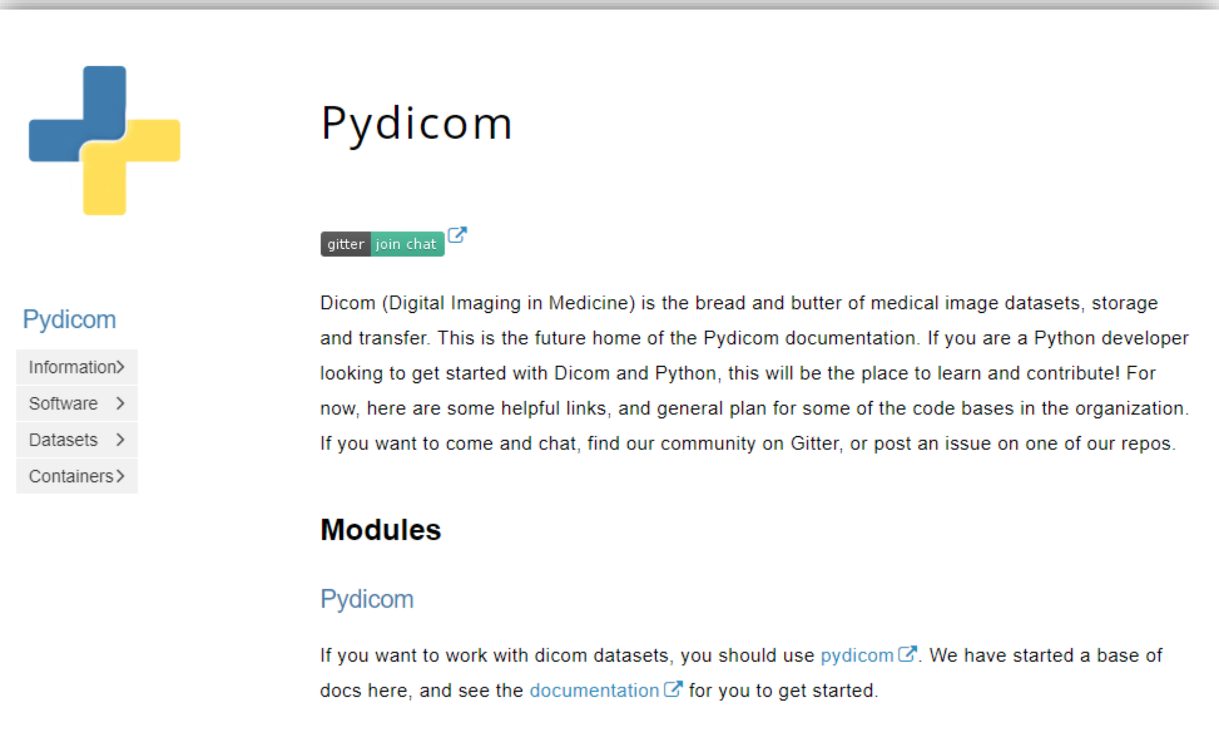
512X512



```
Dataset.file_meta ———————————————————————————————
(0002, 0000) File Meta Information Group Length  UL: 176
(0002, 0001) File Meta Information Version       OB: b'\x00\x01'
(0002, 0002) Media Storage SOP Class UID         UI: CT Image Storage
(0002, 0003) Media Storage SOP Instance UID      UI: 9999.1345638575715201386455629118010580
23921
(0002, 0010) Transfer Syntax UID                 UI: Explicit VR Little Endian
(0002, 0012) Implementation Class UID            UI: 1.2.40.0.13.1.1.1
(0002, 0013) Implementation Version Name         SH: 'dcm4che-1.4.38'
—————————————————————————————————————————————————
(0008, 0018) SOP Instance UID                    UI: ID_7ce5ae372
(0008, 0060) Modality                            CS: 'CT'
(0010, 0020) Patient ID                          LO: 'ID_314d3781'
(0020, 000d) Study Instance UID                  UI: ID_af0d670f5a
(0020, 000e) Series Instance UID                 UI: ID_4195a174b7
(0020, 0010) Study ID                            SH: ''
(0020, 0032) Image Position (Patient)            DS: [-126.408875, -126.408875, -244.165497]
(0020, 0037) Image Orientation (Patient)         DS: [1.000000, 0.000000, 0.000000, 0.00000
0, 1.000000, 0.000000]
(0028, 0002) Samples per Pixel                   US: 1
(0028, 0004) Photometric Interpretation          CS: 'MONOCHROME2'
(0028, 0010) Rows                                US: 512
(0028, 0011) Columns                             US: 512
(0028, 0030) Pixel Spacing                       DS: [0.494750976563, 0.494750976563]
(0028, 0100) Bits Allocated                      US: 16
(0028, 0101) Bits Stored                         US: 16
(0028, 0102) High Bit                            US: 15
(0028, 0103) Pixel Representation                US: 1
(0028, 1050) Window Center                       DS: "35.0"
(0028, 1051) Window Width                        DS: "135.0"
(0028, 1052) Rescale Intercept                   DS: "-1024.0"
(0028, 1053) Rescale Slope                       DS: "1.0"
(7fe0, 0010) Pixel Data                          OW: Array of 524288 elements
```

10434.73MB

Data Format DICOM



Pydicom

gitter | join chat

Dicom (Digital Imaging in Medicine) is the bread and butter of medical image datasets, storage and transfer. This is the future home of the Pydicom documentation. If you are a Python developer looking to get started with Dicom and Python, this will be the place to learn and contribute! For now, here are some helpful links, and general plan for some of the code bases in the organization. If you want to come and chat, find our community on Gitter, or post an issue on one of our repos.

Modules

Pydicom

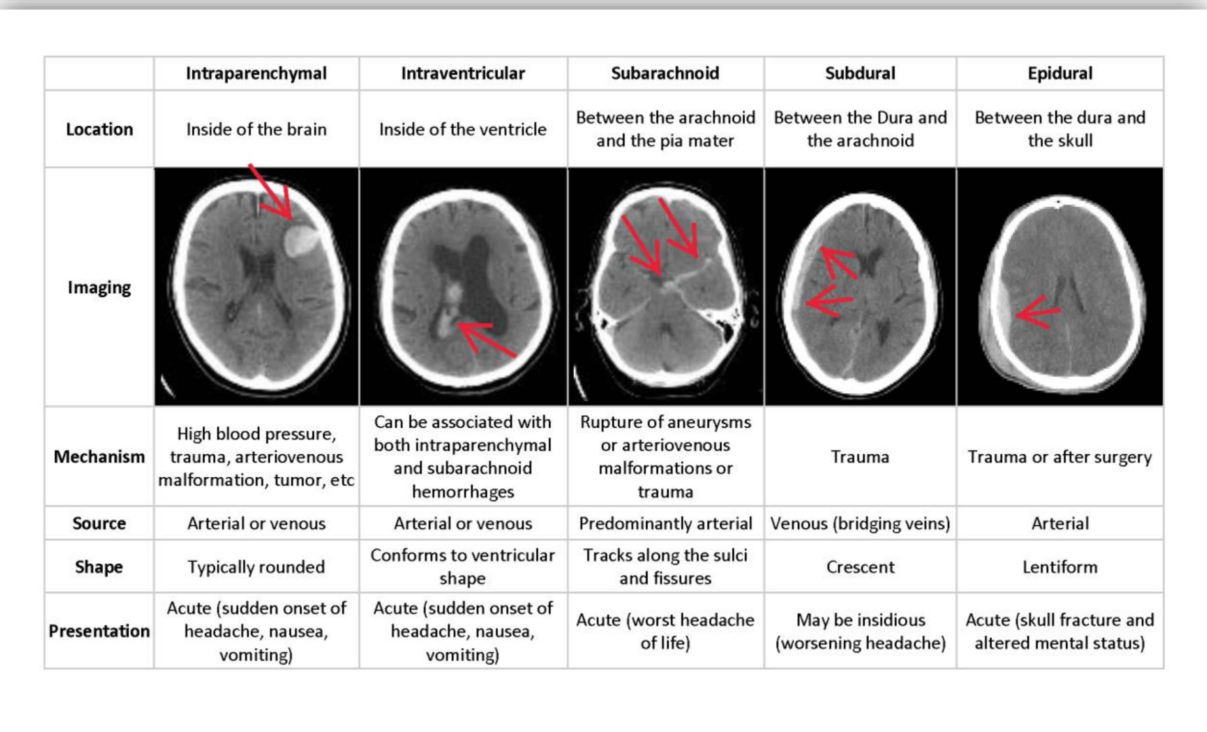
If you want to work with dicom datasets, you should use pydicom. We have started a base of docs here, and see the documentation for you to get started.

DICOM
Digital Imaging and Communications in Medicine
Search
Login
MENU
ABOUT
STANDARD
ACTIVITY
MORE
DICOM makes medical imaging information interoperable
Learn More
DICOM Standard
Working Groups
Presentations

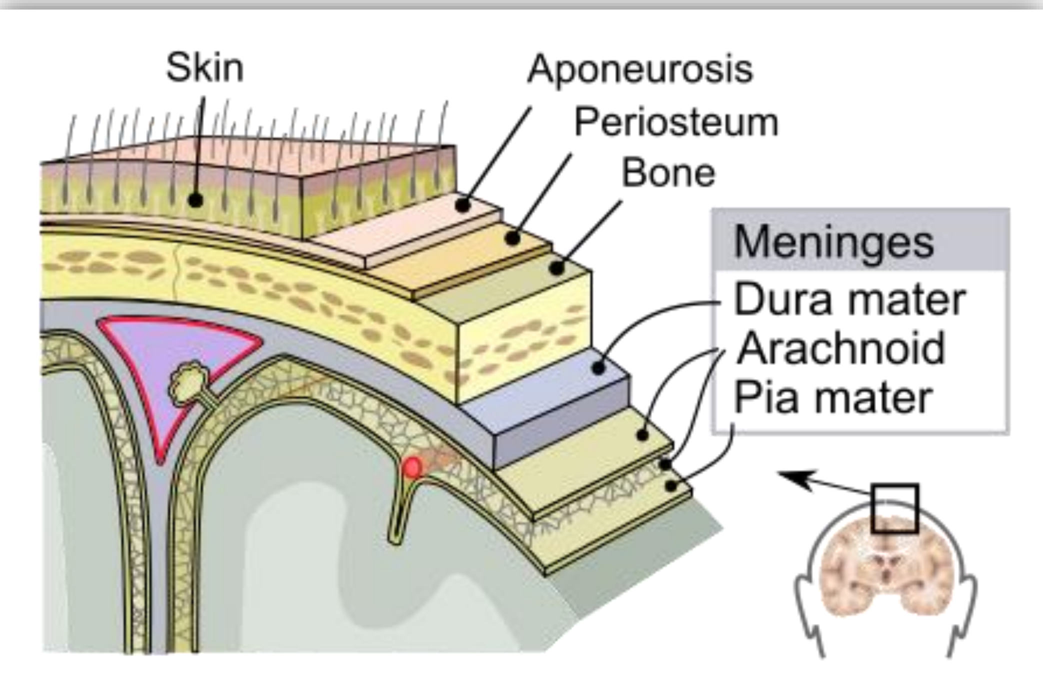
pydicom.github.io

DICOM Digital Imaging and Communications in Medicine www.dicomstandard.org

Classes



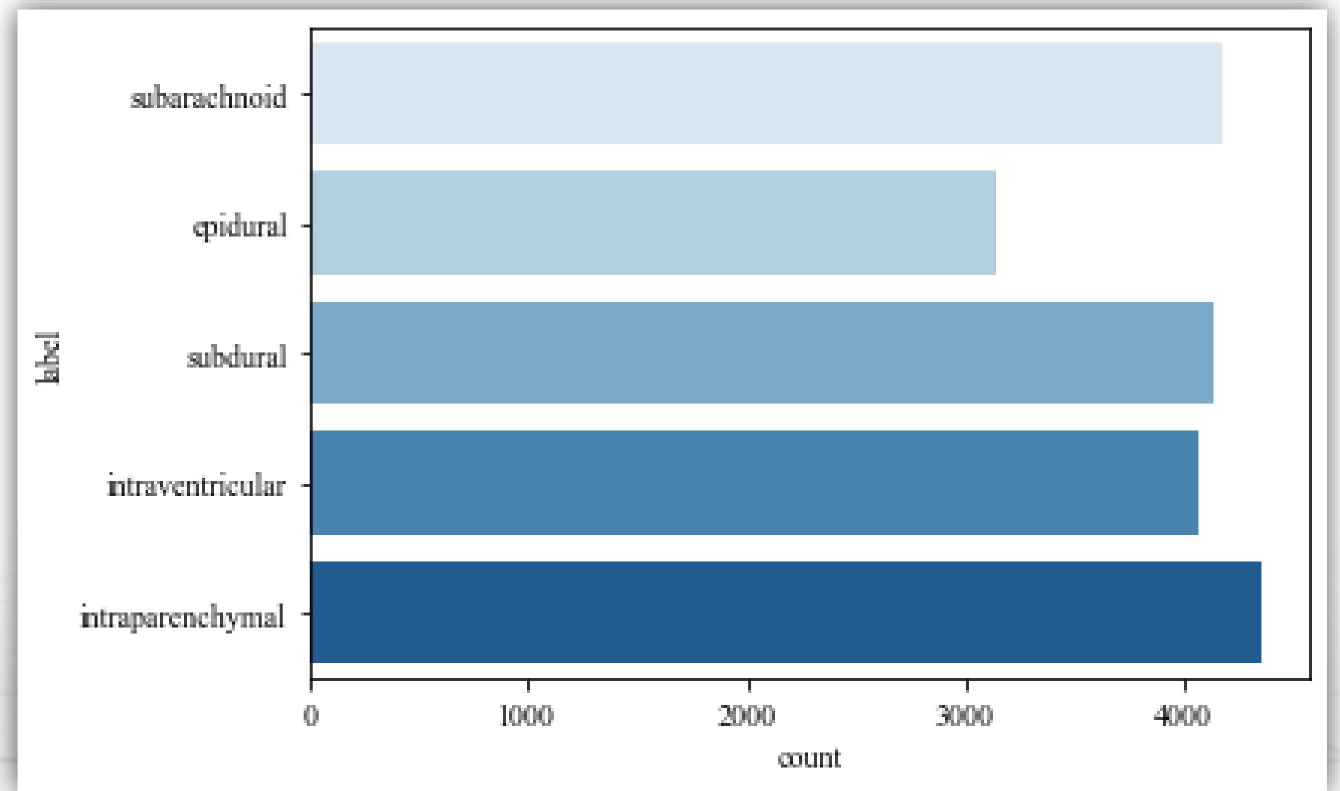
	Intraparenchymal	Intraventricular	Subarachnoid	Subdural	Epidural
Location	Inside of the brain	Inside of the ventricle	Between the arachnoid and the pia mater	Between the Dura and the arachnoid	Between the dura and the skull
Imaging					
Mechanism	High blood pressure, trauma, arteriovenous malformation, tumor, etc	Can be associated with both intraparenchymal and subarachnoid hemorrhages	Rupture of aneurysms or arteriovenous malformations or trauma	Trauma	Trauma or after surgery
Source	Arterial or venous	Arterial or venous	Predominantly arterial	Venous (bridging veins)	Arterial
Shape	Typically rounded	Conforms to ventricular shape	Tracks along the sulci and fissures	Crescent	Lentiform
Presentation	Acute (sudden onset of headache, nausea, vomiting)	Acute (sudden onset of headache, nausea, vomiting)	Acute (worst headache of life)	May be insidious (worsening headache)	Acute (skull fracture and altered mental status)



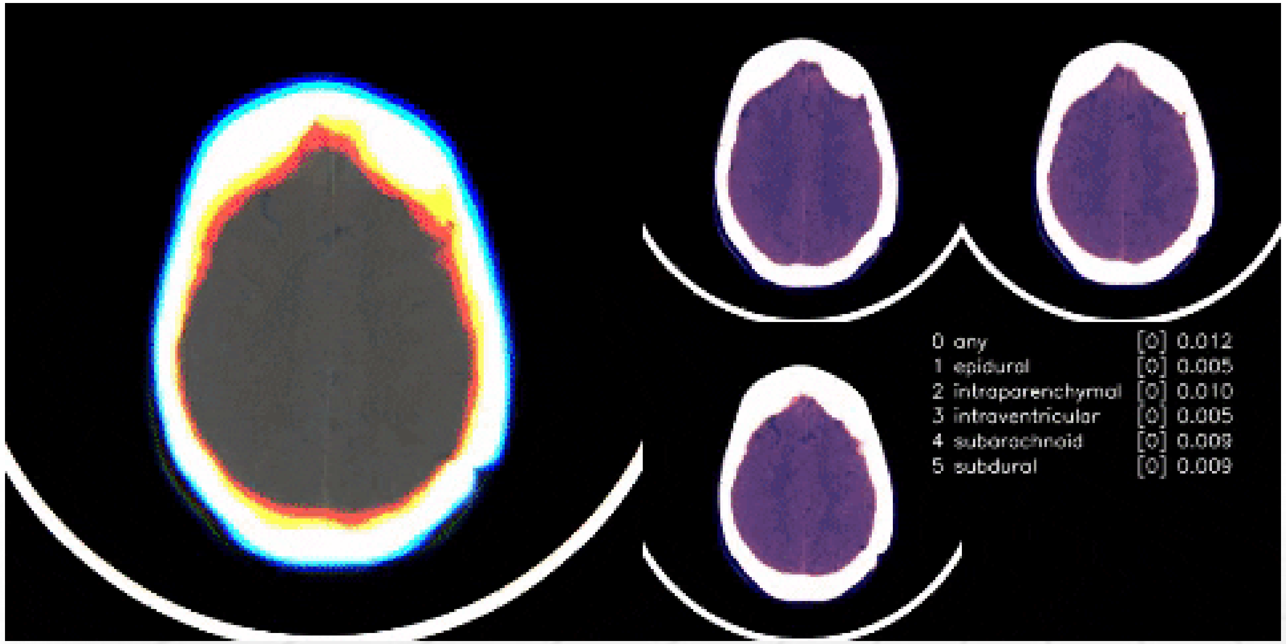
Exploratory Data Analysis

Classes

label	count
subarachnoid	4177
epidural	3145
subdural	4132
intraventricular	4072
intraparenchymal	4351



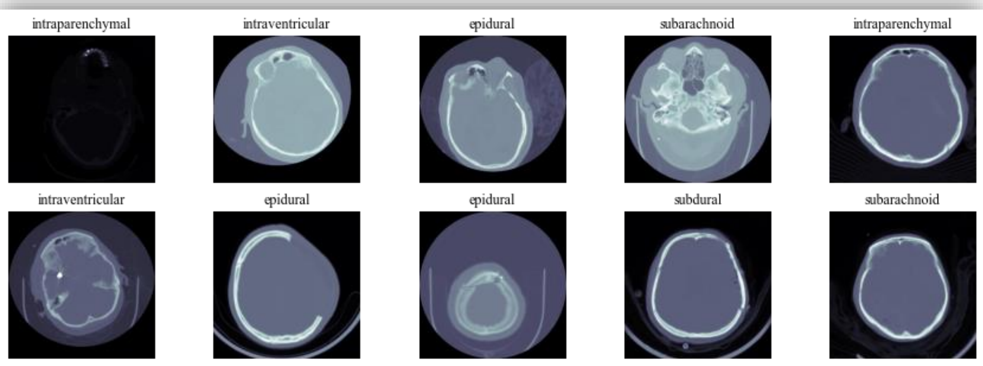
Visualization for Different Windows



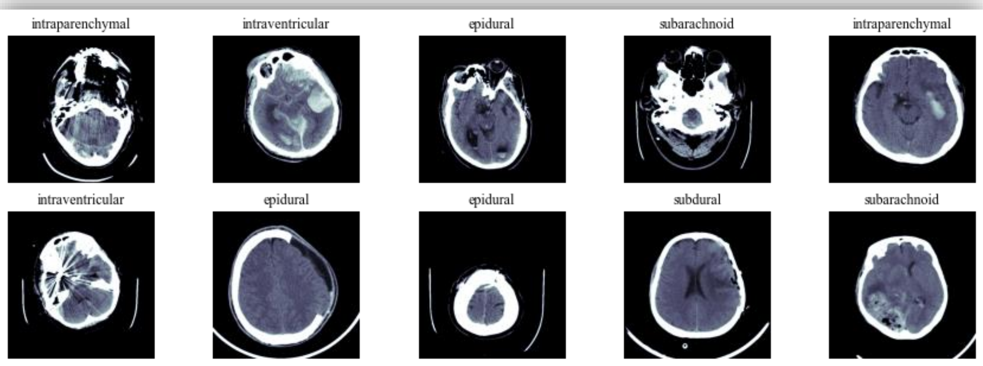
Window:also known as ~grey-level mapping , contrast stretching , histogram modification or contrast enhancement` is the process in which the CT image greyscale component of an image is manipulated via the CT numbers; doing this will change the appearance of the picture to highlight particular structures. The brightness of the image is, adjusted via the window level. The contrast is adjusted via the window width.

Visualization for Different Windows

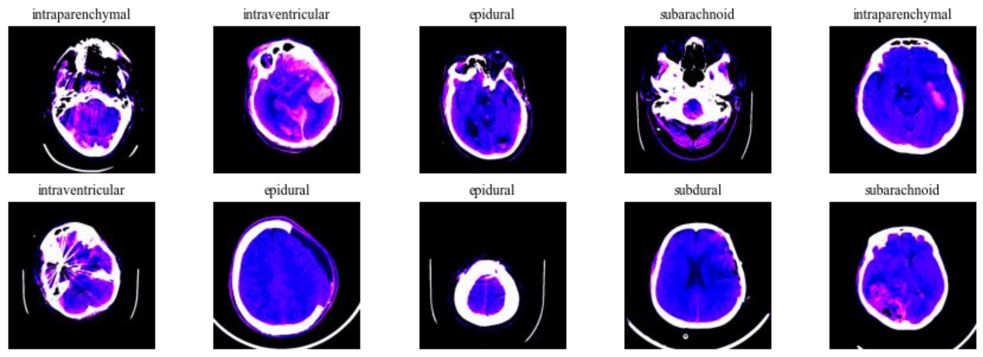
No Windows



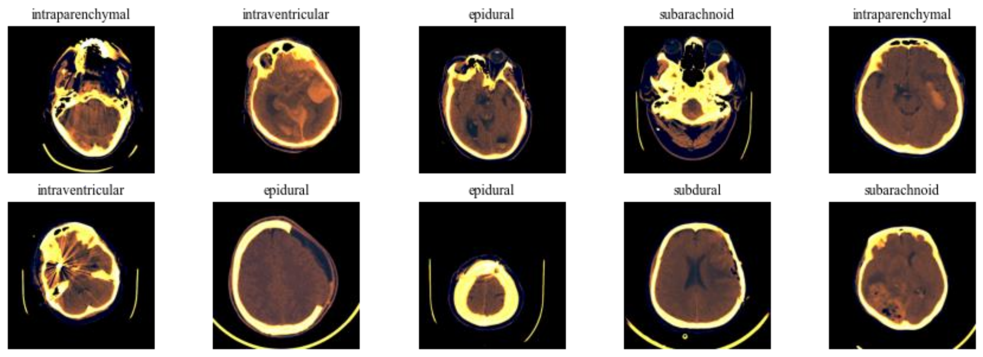
Brain Windows



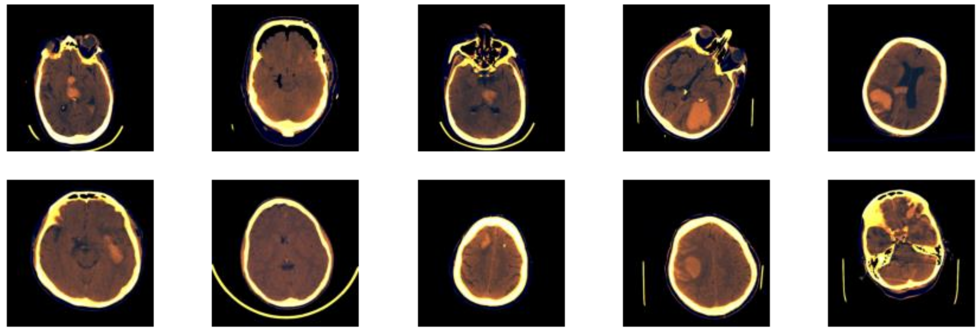
Three Channels Windows



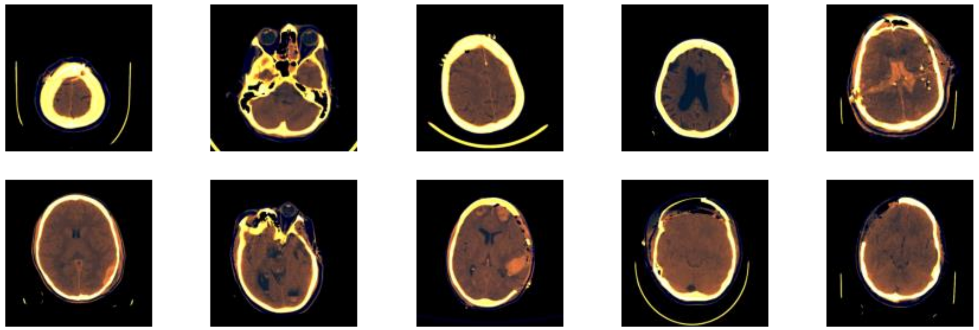
Brain + Subdural + Bone Windows



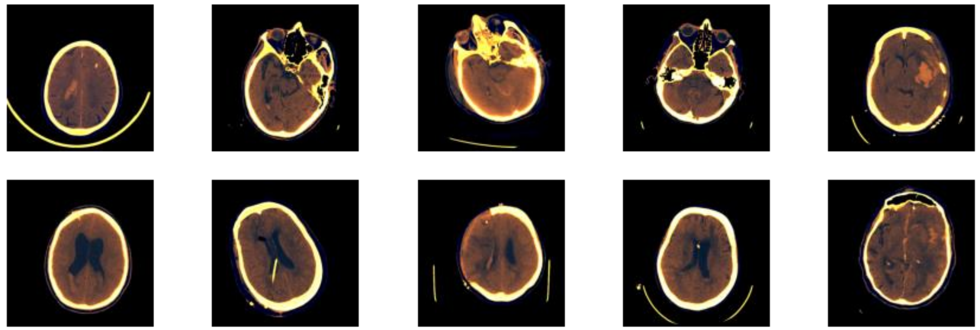
Visualization for Different Classes



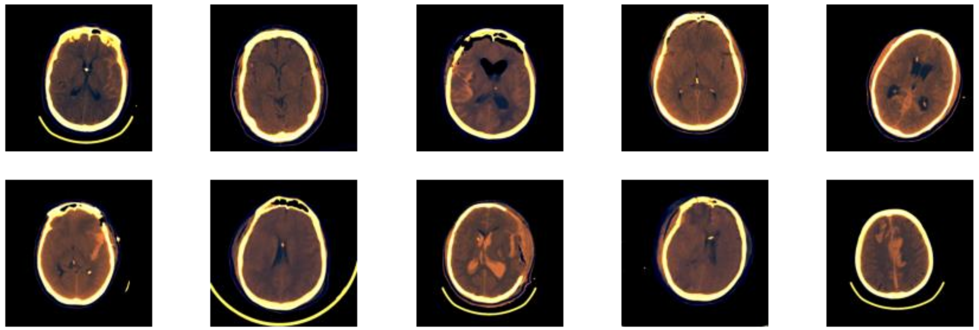
Instances with Intraparenchymal Hemorrhage



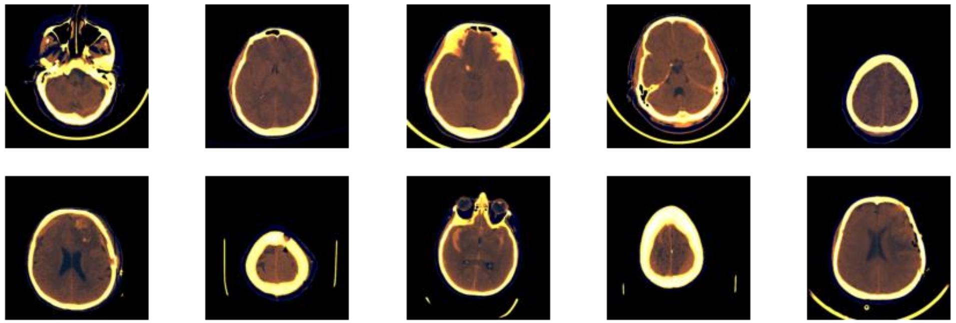
Instances with Epidural Hemorrhage



Instances with Intraventricular Hemorrhage



Subarachnoid

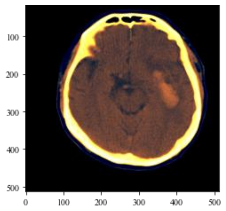


Subdural

Data Preprocess

Data Preprocess

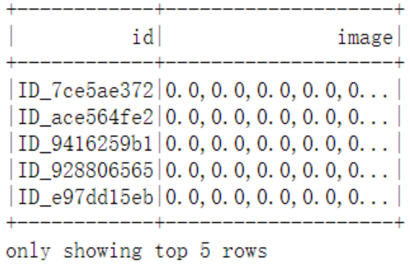
Get windowed image



Rescale image into 299X299

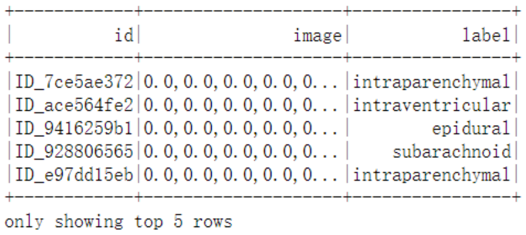
Apply operation by RDD method

Flatten



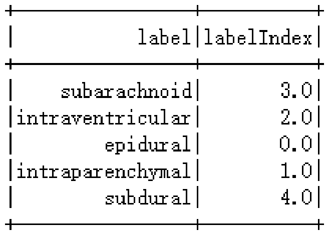
```
+------------+--------------------+
|          id|               image|
+------------+--------------------+
|ID_7ce5ae372|0.0,0.0,0.0,0.0,0...|
|ID_ace564fe2|0.0,0.0,0.0,0.0,0...|
|ID_9416259b1|0.0,0.0,0.0,0.0,0...|
|ID_928806565|0.0,0.0,0.0,0.0,0...|
|ID_e97dd15eb|0.0,0.0,0.0,0.0,0...|
+------------+--------------------+
only showing top 5 rows
```

Add Label



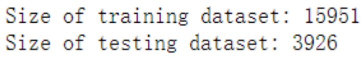
```
+------------+--------------------+----------------+
|          id|               image|           label|
+------------+--------------------+----------------+
|ID_7ce5ae372|0.0,0.0,0.0,0.0,0...|intraparenchymal|
|ID_ace564fe2|0.0,0.0,0.0,0.0,0...|intraventricular|
|ID_9416259b1|0.0,0.0,0.0,0.0,0...|        epidural|
|ID_928806565|0.0,0.0,0.0,0.0,0...|    subarachnoid|
|ID_e97dd15eb|0.0,0.0,0.0,0.0,0...|intraparenchymal|
+------------+--------------------+----------------+
only showing top 5 rows
```

Map Label in to Numeric Value



```
+----------------+----------+
|           label|labelIndex|
+----------------+----------+
|    subarachnoid|       3.0|
|intraventricular|       2.0|
|        epidural|       0.0|
|intraparenchymal|       1.0|
|        subdural|       4.0|
+----------------+----------+
```

Split

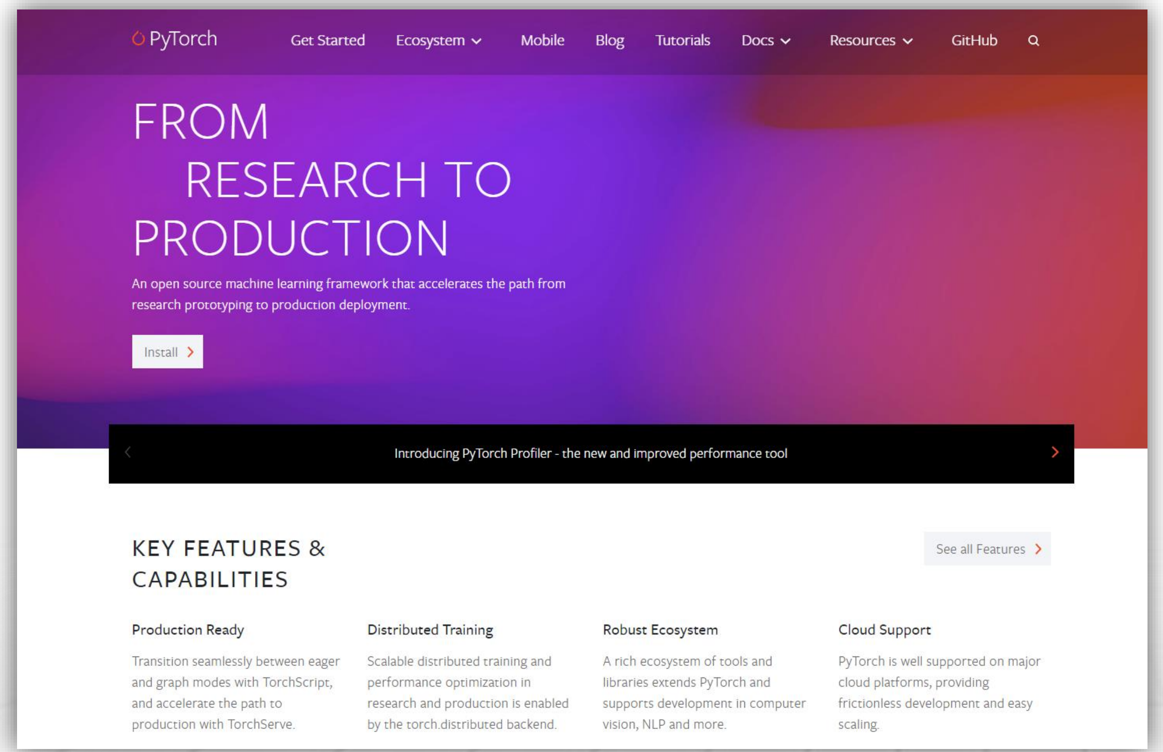


```
Size of training dataset: 15951
Size of testing dataset: 3926
```

Save

Model

Tools



FROM RESEARCH TO PRODUCTION

An open source machine learning framework that accelerates the path from research prototyping to production deployment.

Introducing PyTorch Profiler - the new and improved performance tool

KEY FEATURES & CAPABILITIES

Production Ready

Transition seamlessly between eager and graph modes with TorchScript, and accelerate the path to production with TorchServe.

Distributed Training

Scalable distributed training and performance optimization in research and production is enabled by the torch.distributed backend.

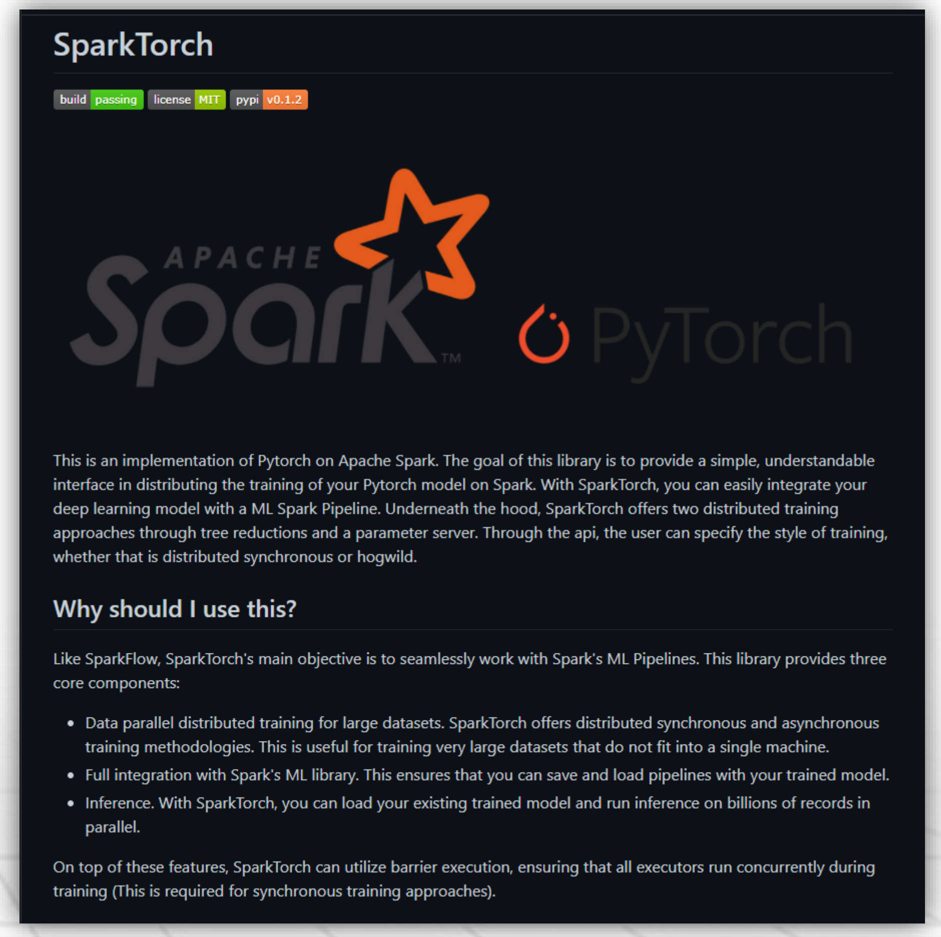
Robust Ecosystem

A rich ecosystem of tools and libraries extends PyTorch and supports development in computer vision, NLP and more.

Cloud Support

PyTorch is well supported on major cloud platforms, providing frictionless development and easy scaling.

PyTorch pytorch.org



SparkTorch

This is an implementation of Pytorch on Apache Spark. The goal of this library is to provide a simple, understandable interface in distributing the training of your Pytorch model on Spark. With SparkTorch, you can easily integrate your deep learning model with a ML Spark Pipeline. Underneath the hood, SparkTorch offers two distributed training approaches through tree reductions and a parameter server. Through the api, the user can specify the style of training, whether that is distributed synchronous or hogwild.

Why should I use this?

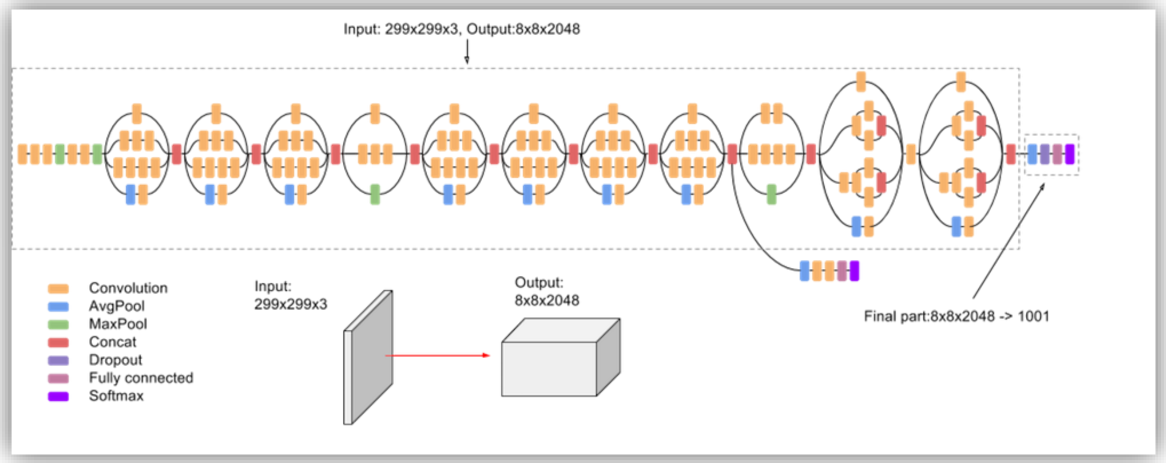
Like SparkFlow, SparkTorch's main objective is to seamlessly work with Spark's ML Pipelines. This library provides three core components:

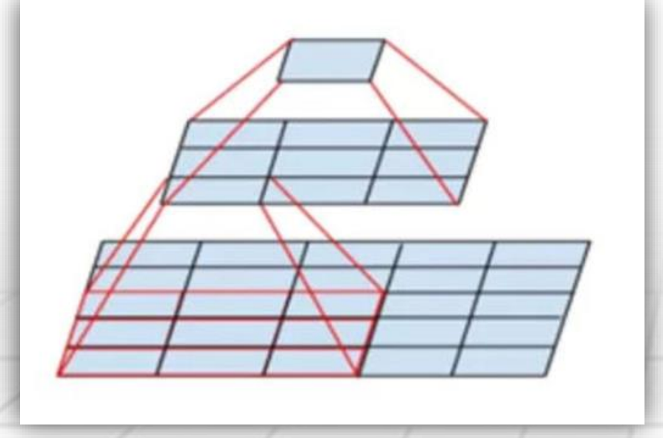
- Data parallel distributed training for large datasets. SparkTorch offers distributed synchronous and asynchronous training methodologies. This is useful for training very large datasets that do not fit into a single machine.
- Full integration with Spark's ML library. This ensures that you can save and load pipelines with your trained model.
- Inference. With SparkTorch, you can load your existing trained model and run inference on billions of records in parallel.

On top of these features, SparkTorch can utilize barrier execution, ensuring that all executors run concurrently during training (This is required for synchronous training approaches).

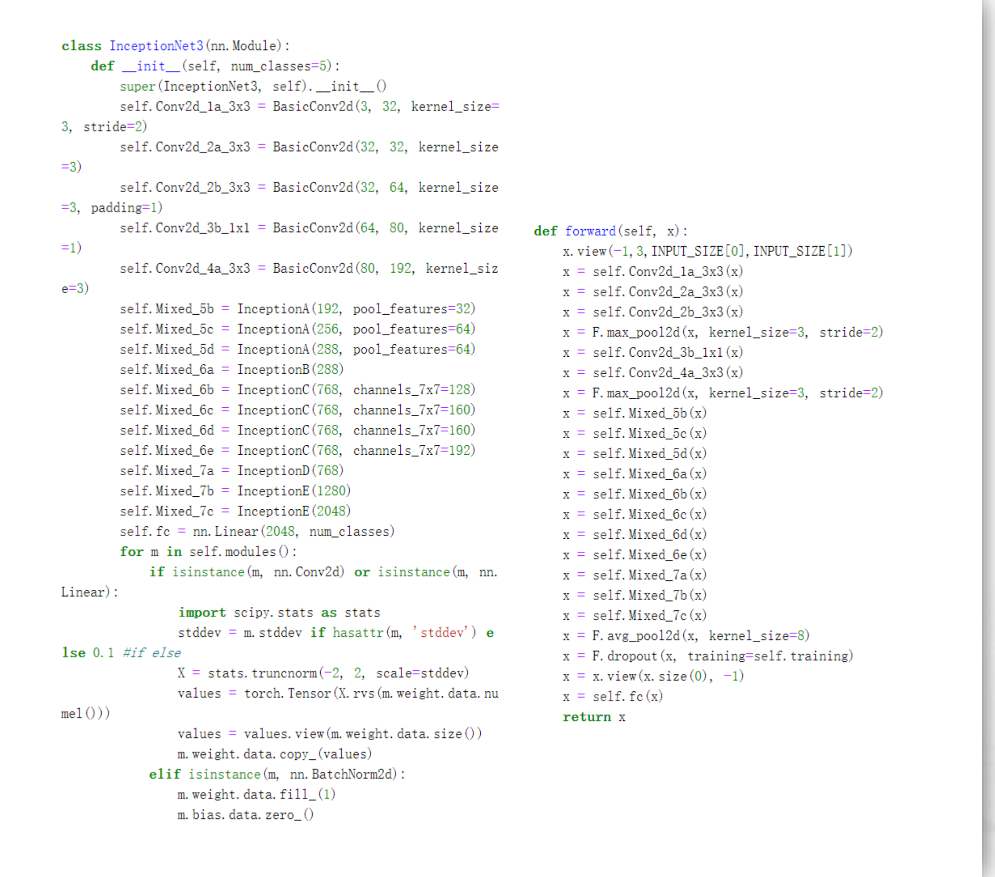
Spark PyTorch github.com/dmmiller612/sparktorch

InceptionNetV3





Model Structure

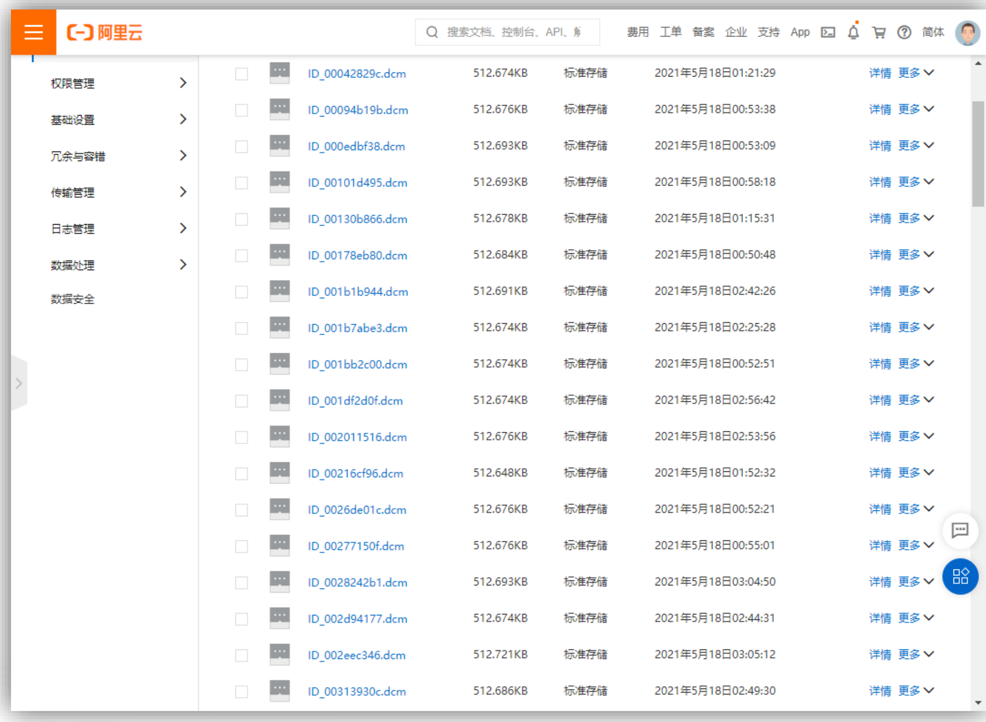


```
class InceptionNet3(nn.Module):
    def __init__(self, num_classes=5):
        super(InceptionNet3, self).__init__()
        self.Conv2d_1a_3x3 = BasicConv2d(3, 32, kernel_size=
3, stride=2)
        self.Conv2d_2a_3x3 = BasicConv2d(32, 32, kernel_size
=3)
        self.Conv2d_2b_3x3 = BasicConv2d(32, 64, kernel_size
=3, padding=1)
        self.Conv2d_3b_1x1 = BasicConv2d(64, 80, kernel_size
=1)
        self.Conv2d_4a_3x3 = BasicConv2d(80, 192, kernel_siz
e=3)
        self.Mixed_5b = InceptionA(192, pool_features=32)
        self.Mixed_5c = InceptionA(256, pool_features=64)
        self.Mixed_5d = InceptionA(288, pool_features=64)
        self.Mixed_6a = InceptionB(288)
        self.Mixed_6b = InceptionC(768, channels_7x7=128)
        self.Mixed_6c = InceptionC(768, channels_7x7=160)
        self.Mixed_6d = InceptionC(768, channels_7x7=160)
        self.Mixed_6e = InceptionC(768, channels_7x7=192)
        self.Mixed_7a = InceptionD(768)
        self.Mixed_7b = InceptionE(1280)
        self.Mixed_7c = InceptionE(2048)
        self.fc = nn.Linear(2048, num_classes)
        for m in self.modules():
            if isinstance(m, nn.Conv2d) or isinstance(m, nn.
Linear):
                import scipy.stats as stats
                stddev = m.stddev if hasattr(m, 'stddev') e
lse 0.1 #if else
                X = stats.truncnorm(-2, 2, scale=stddev)
                values = torch.Tensor(X.rvs(m.weight.data.nu
mel()))
                values = values.view(m.weight.data.size())
                m.weight.data.copy_(values)
            elif isinstance(m, nn.BatchNorm2d):
                m.weight.data.fill_(1)
                m.bias.data.zero_()
```

```
    def forward(self, x):
        x.view(-1,3,INPUT_SIZE[0],INPUT_SIZE[1])
        x = self.Conv2d_1a_3x3(x)
        x = self.Conv2d_2a_3x3(x)
        x = self.Conv2d_2b_3x3(x)
        x = F.max_pool2d(x, kernel_size=3, stride=2)
        x = self.Conv2d_3b_1x1(x)
        x = self.Conv2d_4a_3x3(x)
        x = F.max_pool2d(x, kernel_size=3, stride=2)
        x = self.Mixed_5b(x)
        x = self.Mixed_5c(x)
        x = self.Mixed_5d(x)
        x = self.Mixed_6a(x)
        x = self.Mixed_6b(x)
        x = self.Mixed_6c(x)
        x = self.Mixed_6d(x)
        x = self.Mixed_6e(x)
        x = self.Mixed_7a(x)
        x = self.Mixed_7b(x)
        x = self.Mixed_7c(x)
        x = F.avg_pool2d(x, kernel_size=8)
        x = F.dropout(x, training=self.training)
        x = x.view(x.size(0), -1)
        x = self.fc(x)
        return x
```

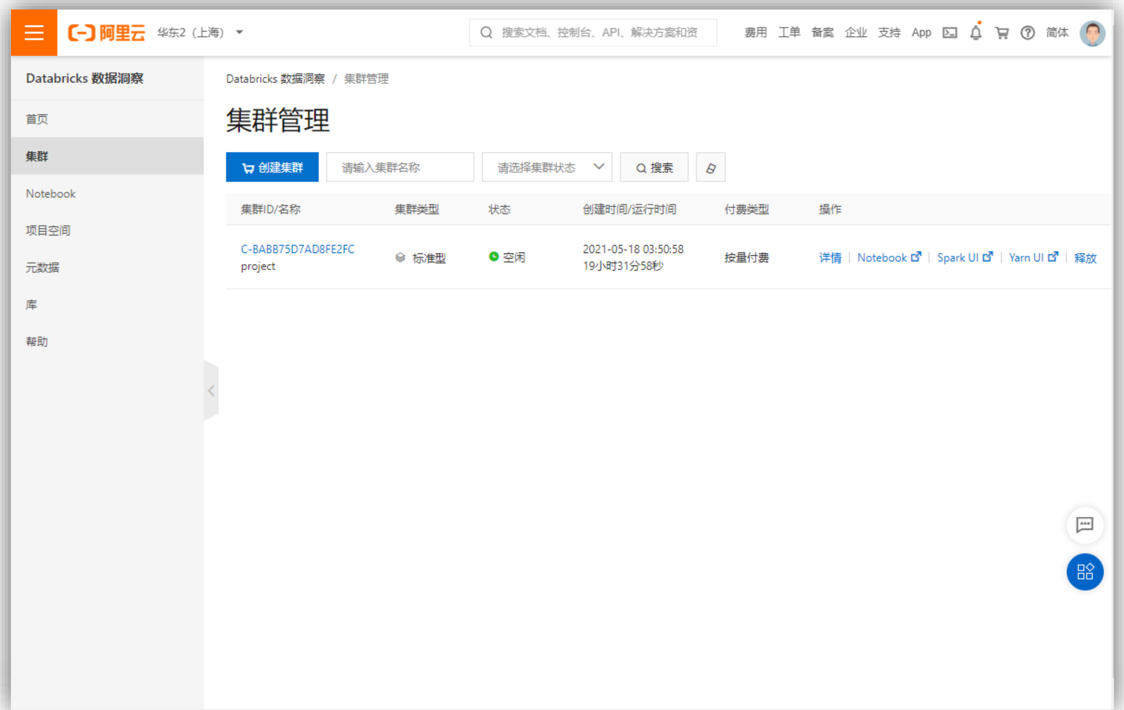
Model Code

Training



ID_00042829c.dcm	512.674KB	标准存储	2021年5月18日01:21:29
ID_00094b19b.dcm	512.676KB	标准存储	2021年5月18日00:53:38
ID_000edbf38.dcm	512.693KB	标准存储	2021年5月18日00:53:09
ID_00101d495.dcm	512.693KB	标准存储	2021年5月18日00:58:18
ID_00130b866.dcm	512.678KB	标准存储	2021年5月18日01:15:31
ID_00178eb80.dcm	512.684KB	标准存储	2021年5月18日00:50:48
ID_001b1b944.dcm	512.691KB	标准存储	2021年5月18日02:42:26
ID_001b7abe3.dcm	512.674KB	标准存储	2021年5月18日02:25:28
ID_001bb2c00.dcm	512.674KB	标准存储	2021年5月18日00:52:51
ID_001df2d0f.dcm	512.674KB	标准存储	2021年5月18日02:56:42
ID_002011516.dcm	512.676KB	标准存储	2021年5月18日02:53:56
ID_00216cf96.dcm	512.648KB	标准存储	2021年5月18日01:52:32
ID_0026de01c.dcm	512.676KB	标准存储	2021年5月18日00:52:21
ID_00277150f.dcm	512.676KB	标准存储	2021年5月18日00:55:01
ID_0028242b1.dcm	512.693KB	标准存储	2021年5月18日03:04:50
ID_002d94177.dcm	512.674KB	标准存储	2021年5月18日02:44:31
ID_002eec346.dcm	512.721KB	标准存储	2021年5月18日03:05:12
ID_00313930c.dcm	512.686KB	标准存储	2021年5月18日02:49:30

OSS

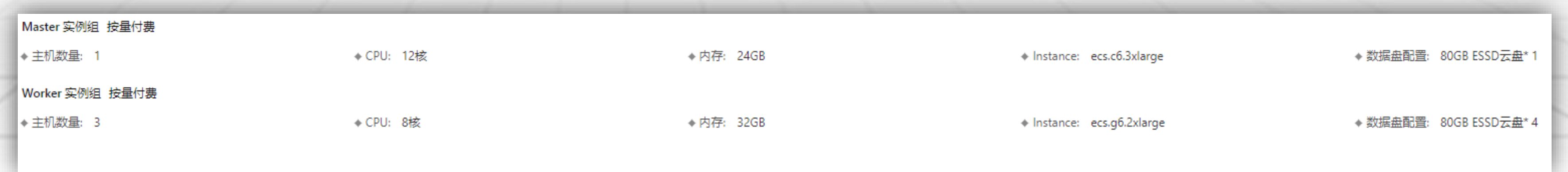


Databricks 数据洞察 / 集群管理

集群管理

集群ID/名称	集群类型	状态	创建时间/运行时间	付费类型	操作
C-BABB75D7AD8FE2FC project	标准型	空闲	2021-05-18 03:50:58 19小时31分58秒	按量付费	详情 | Notebook | Spark UI | Yarn UI | 释放

Databricks



Master 实例组 按量付费

◆ 主机数量: 1 ◆ CPU: 12核 ◆ 内存: 24GB ◆ Instance: ecs.c6.3xlarge ◆ 数据盘配置: 80GB ESSD云盘* 1

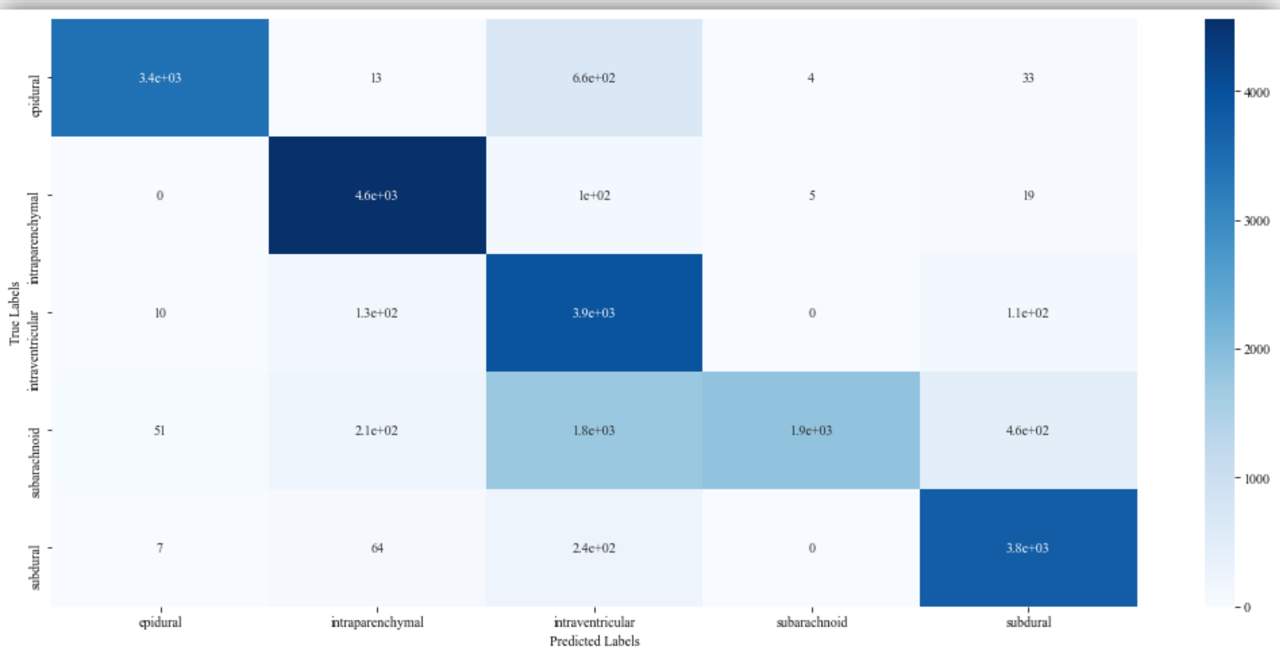
Worker 实例组 按量付费

◆ 主机数量: 3 ◆ CPU: 8核 ◆ 内存: 32GB ◆ Instance: ecs.g6.2xlarge ◆ 数据盘配置: 80GB ESSD云盘* 4

阿里云 www.aliyun.com

Evaluation

Evaluation



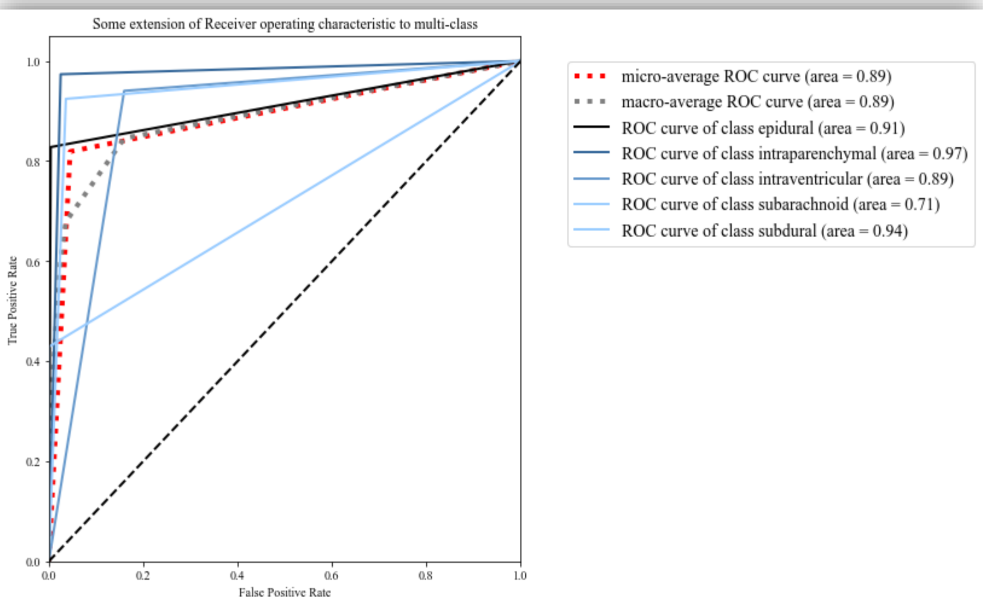
True Labels \ Predicted Labels	epidural	intraparenchymal	intraventricular	subarachnoid	subdural
epidural	3.4e+03	13	6.6e+02	4	33
intraparenchymal	0	4.6e+03	1e+02	5	19
intraventricular	10	1.3e+02	3.9e+03	0	1.1e+02
subarachnoid	51	2.1e+02	1.8e+03	1.9e+03	4.6e+02
subdural	7	64	2.4e+02	0	3.8e+03

Confusion Matrix

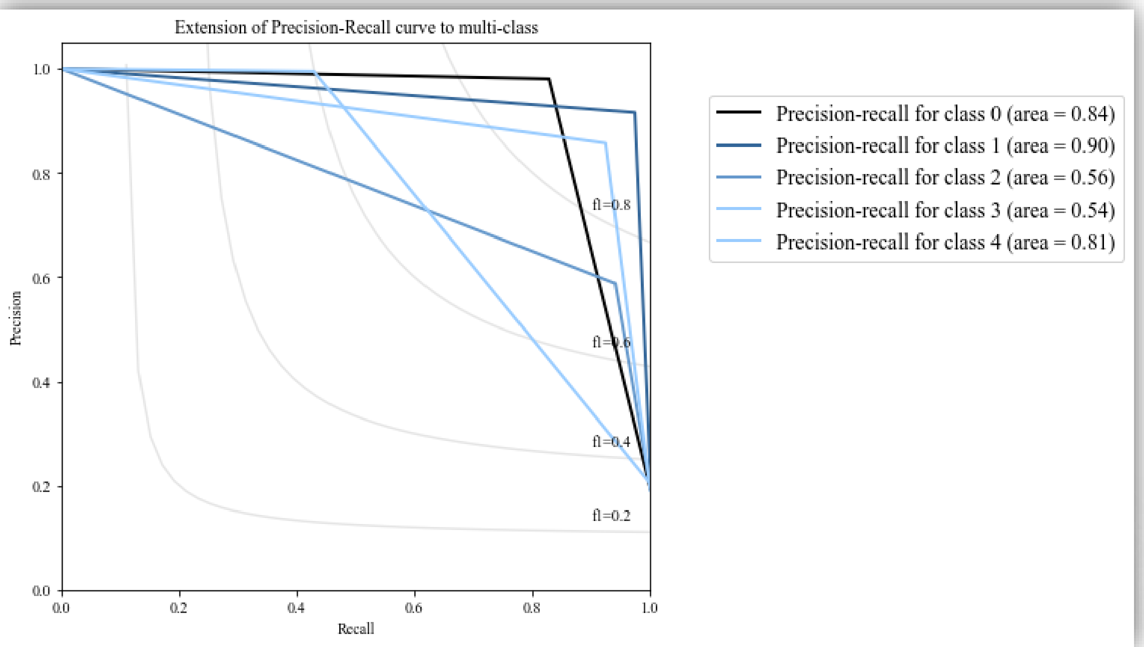
	Score
F1-Score	0.811718
Precision	0.86934
Recall	0.819014
Accuracy	0.819014

Metrics

Evaluation



ROC Curve



PR Curve

Thank for Listening!