包图





MLView.DatasetManagement

数据集管理界面的视图类。

属性：

无

方法：

+ display():void

初次显示界面

+displayCurrentDataset(dataset: MLBLL.Dataset):void

根据一个数据集对象来显示当前上传数据集的所有数据的文件名、标注。

+displayAllDataset(allDataset : ArrayList<MLBLL.Dataset>):void

根据一个数据集数组来显示数据库中所有数据集的信息。



MLView.ModelTrainng

模型训练界面的视图类。

属性：

无

方法：

+ display():void

初次显示界面

+displayAllDataset(allDataset : ArrayList<MLBLL.Dataset>):void

根据一个数据集数组来显示数据库中所有数据集的信息。

+displayTrainingPerformance(testError : float, trainingError: float) : void

根据训练误差、测试误差来简要显示训练时能获得的性能参数。



MLView.DatasettableGenerator

用于生成数据集表格的工具类，用单例模式。

属性：

无

方法：

+getInstance() : DatasetTableGenerator

获取唯一一个类对象。

+getDatasetTable(allDataset : ArrayList<MLBLL.Dataset>):void

根据一个数据集数组在界面上生成一个表格。





MLController.DatabaseManagement.SelectDatasetController

用户点击“选择数据集”的控制器类

属性：

-view : MLView.Datasetmanagement

用于访问数据集管理页面

-datasetManager: MLBLL.DatasetManager

用于调用数据集管理器的操作

方法：

+selectDatasetDescriptionFile(filename:string) : void

用户在界面中选择了数据集描述文件后，此函数会根据该文件名进行后续操作。



MLController.DatabaseManagement.UploadDatasetController

用户点击“上传数据集”的控制器类

属性：

-view : MLView.Datasetmanagement

用于访问数据集管理页面

-datasetManager: MLBLL.DatasetManager

用于调用数据集管理器的操作

方法：

+uploadDataset():void

用户在界面中点击上传数据集后，暂时存在内存中的数据集会上传到数据库。



MLController.DatabaseManagement.CheckDatasetController

用户点击“查看数据集”的控制器类

属性：

-view : MLView.Datasetmanagement

用于访问数据集管理页面

-datasetManager: MLBLL.DatasetManager

用于调用数据集管理器的操作

方法：

+checkDataset():void

用户在界面中点击查看数据集后的操作。





MLController.ModelTraining.CheckDatasetController

用户点击“查看数据集”的控制器类

属性：

-view : MLView.Datasetmanagement

用于访问数据集管理页面

-datasetManager: MLBLL.DatasetManager

用于调用数据集管理器的操作

方法：

+checkDataset():void

用户在界面中点击查看数据集后的操作。



MLController.ModelTraining.TrainingController

用户点击“查看数据集”的控制器类

属性：

-view : MLView.Datasetmanagement

用于访问数据集管理页面

-model : MLBLL.Model

用于调用模型

方法：

+train(

dataset: MLBLL.Dataset,

learningRate: float,

networdDepth: int ) : void

用户在界面上选择好数据集、学习率、网络深度后，点击训练按钮后进行的操作。







MLBLL.Model

用于表示模型，可以进行模型的训练和推理，并返回一些训练时的性能参数。

属性：

- parameterFilename: string

tensorflow模型参数文件的文件名，用于保存模型。

- learningRate : float

超参数，表示学习率，需要在训练前设置。

- networkDepth : int

超参数，表示网络深度，需要在训练前设置。

- trainingError : vector<float>

训练时得到的性能参数，表示四个分类的训练误差。

- testError : vector<float>

训练时得到的性能参数，表示四个分类的测试集误差。

- epochCount : int

训练时得到的性能参数，表示实际迭代次数。

- traingTime : int

训练时得到的性能参数，表示训练误差。

- trainingDataset : PictureDataset

储存训练时的训练数据集。

- testDataset : PictureDataset

储存训练时的测试数据集。

- preprocessInterface : ModelRequiredInterface

保存了一个可以进行图片预处理的接口。

方法：

+ Model()

初始化模型类的所有成员变量。

+ setDataset(dataset : PictureDataset)

根据传入的测试设置测试集。

+ setHyperparameters(learningRate: float, int networkDepth)

根据传入的超参数设置两个超参数。

+ getTestError() : vector<float>

返回四个分类的测试误差。

+ getTraingError() : vector<float>

返回四个分类的训练误差。

+ getEpochCount() : int

返回实际迭代次数。

+ getTrainingTime() : int

返回训练时间秒数。

+ train(int epoch, float precision) : void

传入一个迭代次数和一个准确率要求。在模型超参数和数据集都设置好的情况下，模型会反复训练，直到达到迭代次数或者达到准确率要求。

+ inference(picture : Picture) : TongueLabel

模型对一幅图片进行推理，返回一个舌象标签类。







MLBLL.DatasetManager

数据集的管理类

属性：

- preprocessingInterface : PreprocessingInterface

预处理器的接口类

-dataset : Dataset

缓存的数据集

-datasetDAO : MLDAO.Dataset

访问数据集的数据持久层对象

方法：

+getAllDataset() : ArrayList<Dataset>

获取所有数据集

+ uploadDataset() : void

上传当前缓存的数据集

+ selectDataset(dataset : Dataset) : void

临时保存用户选择的数据集





MLBLL.Dataset

属性：

- pictureData : ArrayList<PictureData>

图片数据数组

+ creationDate : int

用从1900年开始的秒数保存时间

方法：

+ randomGetData(dataCount: int) : vector<PictureData>

根据数据量数随机获取一批数据

+ getAllData() : ArrayList<PictureData>

获取所有数据

+ divide(proportion : float) : Pair<PictureDataset, PictureDataset>

根据比例把数据集拆分成两个数据集



MLBLL.PictureData

表示图片数据和图片标记的集合

属性：

- data : Picture

图片数据对象

- label : TongueLabel

图片标注对象

方法：

+ PictureData()

构造函数。

+ loadData(filename : string) : void

从文件中读取图片。

+ getData() : Picture

获取图片数据。

+ getLabel() : TongueLabel

获取标注数据。



MLBLL.Picture

图片类，用于储存图片的像素信息。

属性：

- arr : vector<int>

储存每个像素的信息，每个像素的颜色用32位表示。

- width : int

图片的宽度。

- height : int

图片的高度。

方法：

+ Picture(filename: string)

通过文件名来构造一个图片对象。

+ getData() : vector<int>

获取图片数据。

+ getWidth() : int

获取宽度。

+ getHeight() : int

获取高度。



MLBLL.TongueLabel

表示舌象的4个标注。

属性：

- coatingColor : int

苔色

- tongueColor : int

舌色

- tongueShape : int

舌形

- tongueStatus : int

舌态

方法：

+ TongueLabel(coatingColor : int, tongueColor : int, tongueShape : int, tongueStatus : int)

用4中属性构造图像标注类。

+ getCoatingColor() : int

获取苔色。

+ getTongueColor() : int

获取舌色。

+ getTongueShape() : int

获取舌型。

+ getTongueStatus() : int

获取舌态。





MLDAO.Dataset

属性：

方法：

+getAllDataset() : ArrayList<Dataset>

从数据库中读取所有数据集表

+ uploadDataset(dataset : Dataset) : void

向数据库上传一个新数据集

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 说明 | 属性 |
| 图片id | int | 用于唯一标识图片 | PK NOT NULL IDENTITY |
| 图片文件名 | Varchar(20) | 图片存储路径，用于访问图片数据 | NOT NULL |
| 舌色 | int | 标注属性 | NOT NULL |
| 苔色 | int | 标注属性 | NOT NULL |
| 舌态 | int | 标注属性 | NOT NULL |
| 舌型 | int | 标注属性 | NOT NULL |
| 数据集id | int | 用于标识数据所属数据集 | FK NOT NULL |

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 说明 | 属性 |
| 数据集id | int | 用于唯一标识一个数据集 | PK NOT NULL IDENTITY |
| 创建时间 | date | 表示数据集的创建时间 |  |