

Chicken-Disease-Classification-Projects

Template.py

```
cnnClassifier/  
├── .github/  
│   └── workflows/  
│       ├── .gitkeep  
│       └── src/  
├── cnnClassifier/  
│   ├── __init__.py  
│   ├── components/  
│   │   ├── __init__.py  
│   │   └── utils/  
│   │       ├── __init__.py  
│   │       └── config/  
│   │           ├── __init__.py  
│   │           └── configuration.py  
│   ├── pipeline/  
│   │   ├── __init__.py  
│   │   └── entity/  
│   │       ├── __init__.py  
│   │       └── constants/  
│   │           ├── __init__.py  
│   │           └── config/  
│   │               ├── config.yaml  
│   │               ├── dvc.yaml  
│   │               ├── params.yaml  
│   │               └── requirements.txt  
│   ├── setup.py  
│   ├── research/  
│   │   └── trials.ipynb  
│   └── templates/  
│       └── index.html
```

Setup.py

Bu Python kodu, setuptools kullanarak bir paket yapılandırması oluşturur. cnnClassifier adlı bir kütüphanenin kurulum bilgilerini belirleyerek, proje açıklaması, yazar bilgileri ve bağlantılar gibi detayları içerir.

Params.yaml

params.yaml dosyası, modelin yapılandırma ayarlarını merkezi bir yerde saklayarak kodun okunabilirliğini artırır ve kolayca değiştirilebilir hale getirir. Bu dosya, görüntü boyutu, batch boyutu, öğrenme hızı gibi önemli parametreleri tanımlayarak modelin eğitim sürecini yönetir.

Config Klasörü

yapılandırma dosyası, bir makine öğrenimi projesinde çeşitli aşamaların dosya yollarını ve parametrelerini belirler. Her bir bölüm, veri alma, model hazırlama, geri çağırımlar ve eğitim gibi süreçlerin çıktılarının saklanacağı dizinleri tanımlar; bu sayede projenin yönetimi ve izlenmesi daha düzenli ve verimli hale gelir.

Src

cnnClassifier

Config

```
class ConfigurationManager  
├── __init__(config_filepath, params_filepath)  
│   ├── self.config = read_yaml(config_filepath)  
│   ├── self.params = read_yaml(params_filepath)  
│   └── create_directories([self.config.artifacts_root])  
  
├── get_data_ingestion_config() -> DataIngestionConfig  
│   ├── config = self.config.data_ingestion  
│   ├── create_directories([config.root_dir])  
│   └── return DataIngestionConfig(...)  
  
├── get_prepare_base_model_config() -> PrepareBaseModelConfig  
│   ├── config = self.config.prepare_base_model  
│   ├── create_directories([config.root_dir])  
│   └── return PrepareBaseModelConfig(...)  
  
├── get_prepare_callback_config() -> PrepareCallbacksConfig  
│   ├── config = self.config.prepare_callbacks  
│   ├── model_ckpt_dir =  
│   │   os.path.dirname(config.checkpoint_model_filepath)  
│   │   ├── create_directories([...])  
│   └── return PrepareCallbacksConfig(...)  
  
├── get_training_config() -> TrainingConfig  
│   ├── training = self.config.training  
├── prepare_base_model = self.config.prepare_base_model  
├── training_data = os.path.join(self.config.data_ingestion.unzip_dir,  
│   │   "Chicken-fecal-images")  
└── create_directories([Path(training.root_dir)])  
    └── return TrainingConfig(...)
```