

**数据结构大作业**

**开发文档**

题目： 基于机器学习的验证码识别算法及实现

2022年6月20日

目录

[一、引言 3](#_Toc107331656)

[1.1 编写目的 3](#_Toc107331657)

[1.2 项目概述 4](#_Toc107331658)

[1.3 项目背景 4](#_Toc107331659)

[1.4 术语和缩略语 4](#_Toc107331660)

[1.5 参考文献 5](#_Toc107331661)

[1.6 项目目标 6](#_Toc107331662)

[二、开发计划 6](#_Toc107331663)

[2.1 最终呈现形式 6](#_Toc107331664)

[2.2 主要功能描述 6](#_Toc107331665)

[2.3 开发环境 7](#_Toc107331666)

[2.4 验收标准 7](#_Toc107331667)

[2.5 进度安排 7](#_Toc107331668)

[2.6 计算公式 8](#_Toc107331669)

[三、可行性分析 8](#_Toc107331670)

[3.1 市场可行性分析 8](#_Toc107331671)

[3.2 技术可行性分析 8](#_Toc107331672)

[3.3 资源可行性分析 8](#_Toc107331673)

[四、需求分析 8](#_Toc107331674)

[4.1 数据描述 8](#_Toc107331675)

[4.2 功能需求 9](#_Toc107331676)

[五、概要设计 9](#_Toc107331677)

[5.1 处理流程 9](#_Toc107331678)

[5.2 数据结构设计 10](#_Toc107331679)

[5.3 运行设计 10](#_Toc107331680)

[六、详细设计 13](#_Toc107331681)

[6.1 图像处理 13](#_Toc107331682)

[6.1.1功能描述 13](#_Toc107331683)

[6.1.2输入 13](#_Toc107331684)

[6.1.3输出 14](#_Toc107331685)

[6.1.4关键代码展示 14](#_Toc107331686)

[6.1.5训练结果 16](#_Toc107331687)

[6.2可视化功能模块 17](#_Toc107331688)

[6.2.1 功能描述 17](#_Toc107331689)

[6.2.2模型结构 18](#_Toc107331690)

# 一、引言

## 1.1 编写目的

验证码(captcha)是“completely automated public Turing test to tell computers and humans apart”(全自动区分计算机和人类的图灵测试)的缩写，验证码在设计之初只是一个用来区分人和计算机程序的图灵测试,随着互联网的发展,现在的验证码主要作为维护互联网安全的工具。验证码广泛应用在互联网各个领域,也称为互联网安全防火墙,本质上验证码是为了区分人与机器人,防止网络机器人的攻击。一般会出现在用户登录网站、网站的注册、信息的查询、网站发帖等场景。在这些场景中我们希望用户是一个真正的人,而不是计算机程序。图片字符识别对人类来说是很简单的,然而对计算机程序来说却比较难。因此验证码的存在能一定程度的维护一个良好的网络环境。

全球通行的字母、数字验证码是目前最为广泛使用的一种算法较为简单，相对访客而言比较人性化的简单设计之一。验证码的识别方式主要有两种，一种是将验证码切割为单独的字符后再以机器学习或模板匹配等方法进行识别，另一种是利用神经网络对图片进行整体识别，这两种方法在应对不同的字符验证码识别场景中各有利弊。

验证码的识别结合了图像处理、模式识别等多个领域的研究，对于促进各领域的技术研究有着重要的意义。机器学习是人工智能的重要分支，我们主要通过机器学习使计算机能够简单的识别出验证码。

## 1.2 项目概述

我们具体研究图像类验证码，图像类验证码大多是数字、字母的组合，国内也有使用汉字的。在这个基础上增加噪点、干扰线、变形、重叠、不同字体颜色等方法来增加识别难度。相应的，我们的验证码识别大体可以分为下面几个方面：灰度处理、增加对比度（增加色差）、二值化、降噪、倾斜校正分割字符、建立训练库、识别等。我们主要运用算法，构建KNN的分类器。将各个数字和字母分类装载并进行训练。到最后通过机器学习我们上传一张验证码就可以得到它的正确数字。

我们将字符分为简单字符，带干扰的字符，简单的计算字符等进行分类，方便进行快速获得。我们也打算进一步识别不同验证码。

## 1.3 项目背景

在当前网络环境中，验证码本身是用来区分人机的一种全自动验证程序，在一定程度上提高了网络防御力，又被称为“反向图灵测试”。目前，多数网站在用户注册、登录、下载甚至发表评论等情况下都用验证码进行验证，从而保证合法有效的用户权益．一般要求人类的识别率达到90％以上，机器的识别率低于0.01% 。验证码的分类有滑块验证码、文本验证码、图像验证码、拼图验证码等。我们主要研究图像验证码。

## 1.4 术语和缩略语

[1] 验证码：

验证码（CAPTCHA）是“Completely Automated Public Turing test to tell Computers and Humans Apart”（全自动区分计算机和人类的图灵测试）的缩写，是一种区分用户是计算机还是人的公共全自动程序。可以防止：恶意破解密码、刷票、论坛灌水，有效防止某个黑客对某一个特定注册用户用特定程序暴力破解方式进行不断的登陆尝试，实际上用验证码是现在很多网站通行的方式，我们利用比较简易的方式实现了这个功能。这个问题可以由计算机生成并评判，但是必须只有人类才能解答。由于计算机无法解答CAPTCHA的问题，所以回答出问题的用户就可以被认为是人类。

[2] CNN:

卷积神经网络（Convolutional Neural Networks, CNN）是一类包含卷积计算且具有深度结构的前馈神经网络，是深度学习的代表算法之一。卷积神经网络具有表征学习能力，能够按其阶层结构对输入信息进行平移不变分类，因此也被称为“平移不变人工神经网络”。

对卷积神经网络的研究始于二十世纪80至90年代，时间延迟网络和LeNet-5是最早出现的卷积神经网络；在二十一世纪后，随着深度学习理论的提出和数值计算设备的改进，卷积神经网络得到了快速发展，并被应用于计算机视觉、自然语言处理等领域。

卷积神经网络仿造生物的视知觉机制构建，可以进行监督学习和非监督学习，其隐含层内的卷积核参数共享和层间连接的稀疏性使得卷积神经网络能够以较小的计算量对格点化特征，例如像素和音频进行学习、有稳定的效果且对数据没有额外的特征工程要求。

[3] KNN：

KNN（K- Nearest Neighbor）法即K最邻近法，最初由 Cover和Hart于1968年提出，是一个理论上比较成熟的方法，也是最简单的机器学习算法之一。该方法的思路非常简单直观：如果一个样本在特征空间中的K个最相似（即特征空间中最邻近）的样本中的大多数属于某一个类别，则该样本也属于这个类别。该方法在定类决策上只依据最邻近的一个或者几个样本的类别来决定待分样本所属的类别。

## 1.5 参考文献

1. 基于机器学习的系统登录验证码自动识别与安全风险研究 吴俪昊、张晓曙、梁雪枫 [基于机器学习的系统登录验证码自动识别与安全风险研究 (cqvip.com)](http://qikan.cqvip.com/Qikan/Article/ReadIndex?id=7102501444&info=5ZaTbi5lSoCaF%2b5%2b4ryBBRD3aW5Nm76UL5A0VJ6%2fRYH34w0BSuXzVA%3d%3d)
2. 基于机器学习的验证码识别 侍啸，李佳霖 [基于机器学习的验证码识别.ashx (wanfangdata.com.cn)](https://oss.wanfangdata.com.cn/www/%E5%9F%BA%E4%BA%8E%E6%9C%BA%E5%99%A8%E5%AD%A6%E4%B9%A0%E7%9A%84%E9%AA%8C%E8%AF%81%E7%A0%81%E8%AF%86%E5%88%AB.ashx?isread=true&type=perio&resourceId=zgzlxxcy201826145&transaction=%7B%22id%22%3Anull%2C%22transferOutAccountsStatus%22%3Anull%2C%22transaction%22%3A%7B%22id%22%3A%221516391518666092544%22%2C%22status%22%3A1%2C%22createDateTime%22%3Anull%2C%22payDateTime%22%3A1650370885899%2C%22authToken%22%3A%22TGT-36644077-eAdLnWgRyXCcASINhTfdnXs9aSQfjOeFNslTToh5zWVCWXRuA5-my.wanfangdata.com.cn%22%2C%22user%22%3A%7B%22accountType%22%3A%22Group%22%2C%22key%22%3A%22sddx%22%7D%2C%22transferIn%22%3A%7B%22accountType%22%3A%22Income%22%2C%22key%22%3A%22PeriodicalFulltext%22%7D%2C%22transferOut%22%3A%7B%22GTimeLimit.sddx%22%3A3.0%7D%2C%22turnover%22%3A3.0%2C%22orderTurnover%22%3A3.0%2C%22productDetail%22%3A%22perio_zgzlxxcy201826145%22%2C%22productTitle%22%3Anull%2C%22userIP%22%3A%22112.6.227.3%22%2C%22organName%22%3Anull%2C%22memo%22%3Anull%2C%22orderUser%22%3A%22sddx%22%2C%22orderChannel%22%3A%22pc%22%2C%22payTag%22%3A%22%22%2C%22webTransactionRequest%22%3Anull%2C%22signature%22%3A%22grtw2Y%2Fmw8xNMKLJx7y0Y6FxBs7cwRWnSbgBkApA%2Bxw1H%2B3MYD0TxQD8%2FYjrMoviQD%2BT4EP%2Fkh%2BN%5Cn4%2Fk1jfFUheiBKhpWcaT1bsJNC1m0FG0pbG%2Fgpk9FzhNBYrKfp5gQImDwWymbXcFZ42wc79rxw3wJ%5Cnn%2B8WYng5lwxosNHPoU4%3D%22%2C%22delete%22%3Afalse%7D%2C%22isCache%22%3Afalse%7D)
3. 基于机器学习的字符图像验证码识别研究 田勇 [基于机器学习的字符图像验证码识别研究 (cqvip.com)](http://qikan.cqvip.com/Qikan/Article/ReadIndex?id=7104942764&info=b3xZKkMXyTVVzqlp%2fo3EVhz55GuAOc3%2bT0%2bVbsexMAofu5XJ1iUvsg%3d%3d)
4. 基于Python和CNN的数字验证码识别 张国荣，刘炳君，付成丽 [基于Python和CNN的数字验证码识别 - 百度学术 (baidu.com)](https://xueshu.baidu.com/usercenter/paper/show?paperid=1c5b0r40236106x0vr3t0440a3427972)

## 1.6 项目目标

该项目可以实现对简单验证码的识别。帮助不想输入验证码的人群实现自动识别，帮助复制粘贴验证码。验证码识别结合了图像处理、模式识别等多个领域的研究，对于促进各领域的技术研究具有重要意义。机器学习是人工智能的重要分支，即通过计算机网络处理各个变量间的复杂和非线性关系，并使误差最小化的方法。我们可以通过该项目加强对机器学习的认识。

# 二、开发计划

## 2.1 最终呈现形式

以网页形式进行呈现，页面简介。上传一张验证码图像就可以得到验证码文本。

## 2.2 主要功能描述

在网页中上传验证码文件，选择对应的验证码种类，点击确定完成识别。

验证码的种类有简单的四位纯数字验证码，简单四位纯字母验证码，四位字母数字混合编码、简单的数字四则运算。可以根据验证码的不同进行选择，得到符合要求的验证码。

## 2.3 开发环境

开发环境：Python 3.10 tensorflow 2.8.0 sklearn Pillow

运行工具：Pycharm 2021 vscode

## 2.4 验收标准

上传一张验证码照片，点击确定后可以得出正确的验证码。

## 2.5 进度安排

2022年4月-2022年5月：

数据结构的学习

学习机器学习和深度学习的知识

网站搭建基础语法的学习

获取训练数据

2022年5月-2022年6月：

Web网页的开发

模型的搭建

将模型应用于网站

2022年6月-2022年7月：

提交结题报告

## 2.6 计算公式

对于图像中某红、绿、蓝三基色分量分别为R、G、B的像素来说，其灰度值为

Gray=R×0.299+G×0.587+B×0.114，其中Gray表示灰度值。

# 三、可行性分析

## 3.1 市场可行性分析

可以帮助需要验证码的软件开发人员或者网站开发人员提供大量的测试数据，方便进行灌数据，测试功能等。

可以帮助用户更快速有效的识别验证码。

## 3.2 技术可行性分析

当前机器学习技术已经成熟，Python中可以运用大量第三方库，如Pillow，Tensorflow等。

## 3.3 资源可行性分析

小组成员为正在学习Python和Web技术的“技术人员”，电脑配有Python的开发环境及IDE，也具有前端的IDE。框架可以帮助快速搭建网站。

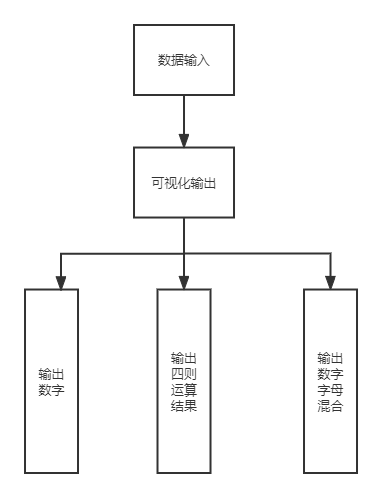
# 四、需求分析

## 4.1 数据描述

采用用户输入的数据，分为简单四位纯数字、简单四位纯字母、数字加字母、数字四则运算四种形式，对于干扰的形式有噪波和数字的扭曲旋转。

## 4.2 功能需求

4.2.1 功能模块结构图



# 五、概要设计

## 5.1 处理流程

1.简单纯数字/字母

A）通过将输入的纯数字或者字母保存到对应的文件中

B)通过cv2进行灰度处理、去噪二值化对图片进行处理

C）切割图片并训练字体库

D）识别

2.混合数字字母

A）通过将输入的纯数字或者字母保存到对应的文件中

B)通过cv2进行灰度处理、去噪二值化对图片进行处理

C）切割图片、倾斜度矫正并训练字体库

D）识别

3.变形四则运算

A）通过将输入的纯数字或者字母保存到对应的文件中

B)通过cv2进行灰度处理、去噪二值化对图片进行处理

C）切割图片、倾斜度矫正并训练字体库

D）识别并进行四则运算

## 5.2 数据结构设计

5.2.1 逻辑结构设计

采用集合结构，线性结构

5.2.2 物理结构设计

采用顺序结构存储

5.2.3数据结构与程序的关系

采用集合结构，如可视化分析的去除重复信息，采用线性结构，存储图片的结果和编号

## 5.3 运行设计

1.原始界面



2.验证码为纯数字时



3.验证码为纯字母时



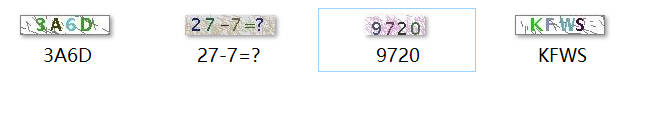
4.验证码为数字和字母的混合形式时



5.验证码为简单的数字混合运算时



6.源数据



# 六、详细设计

## 6.1 图像处理

### 6.1.1功能描述

使用灰度处理，去噪二值化等将图片转换为字符

### 6.1.2输入

不同类型的图片，包括纯数字/字母类型，扭曲数字/字母类型，数字字母混合类型，数字四则运算类型。

### 6.1.3输出

纯数字/字母类型，扭曲数字/字母类型，数字字母混合类型输出验证码数字和字母，数字四则运算类型输出四则运算结果数字

### 6.1.4关键代码展示

**1.色差增强**

def convert\_to\_pure\_black\_white(image):

width = image.shape[1]

height = image.shape[0]

image[0] = 255

for line in image:

line[0] = 255

line[-1] = 255

image[-1] = 255

for w in range(width):

for h in range(height):

if image[h][w] < 237:

image[h][w] = 0

else:

image[h][w] = 255

image2 = image[:, 13:73]

return image2

**2.干扰线清理**

由于进行去噪的过程中还有些较粗的干扰线没有去除，所以还要重新进行干扰线处理

def remove\_noise\_line(image):

width = image.shape[1]

height = image.shape[0]

for w in range(width):

count = 0

for h in range(height):

if image[h][w] < 100:

count += 1

else:

if 2 > count > 0:

for c in range(count):

image[h - c - 1][w] = 255

count = 0

for h in range(height):

count = 0

for w in range(width):

if image[h][w] < 100:

count += 1

else:

if 2 > count > 0:

for c in range(count):

image[h][w - c - 1] = 255

count = 0

return image

**3.构建knn训练器**

def load\_dataset():

X = []

y = []

for i in range(70):

path = "./train/%d%d.png" % (i / 7, i % 7)

pix = np.asarray(Image.open(path).convert("L"))

X.append(pix.reshape(9\*13))

y.append(int(i / 7))

return np.asarray(X), np.asarray(y)

from sklearn.neighbors import KNeighborsClassifier

knn = KNeighborsClassifier()

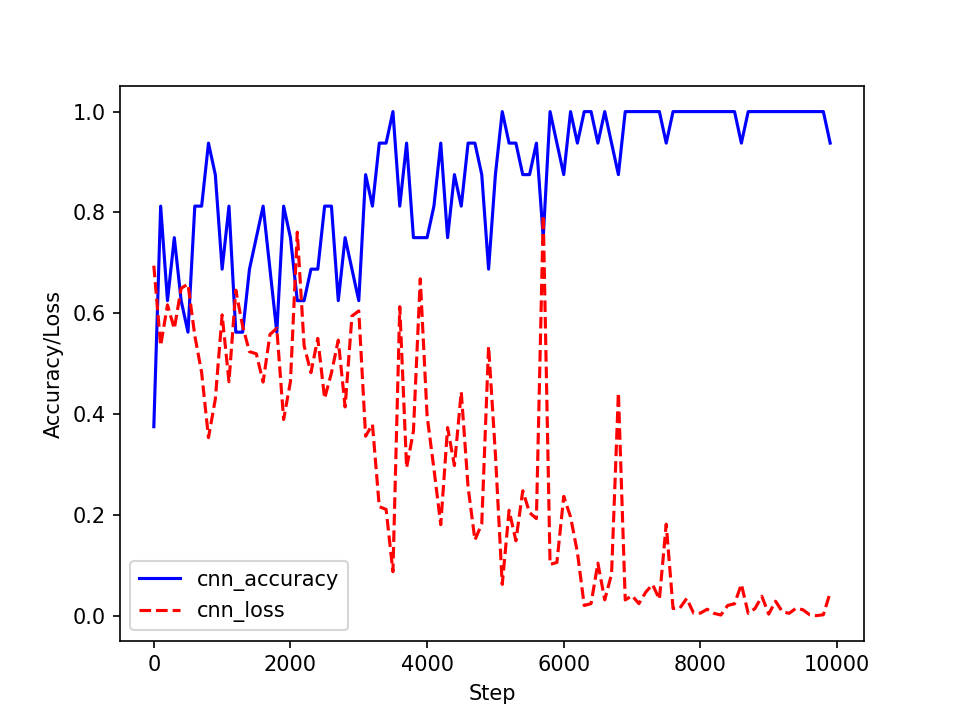
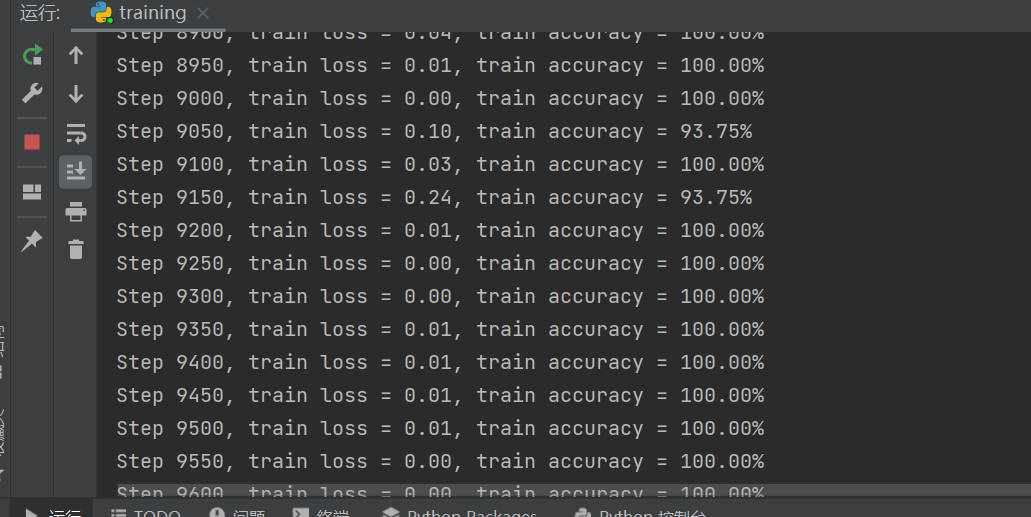
knn.fit(X, y)

**4切割字符**

def split\_letters(image):

letters = [image[:, : 15], image[:, 15: 30], image[:, 30: 45], image[:, 45: 60]]

return letters

6.1.5训练结果

## 6.2可视化功能模块

### 6.2.1 功能描述

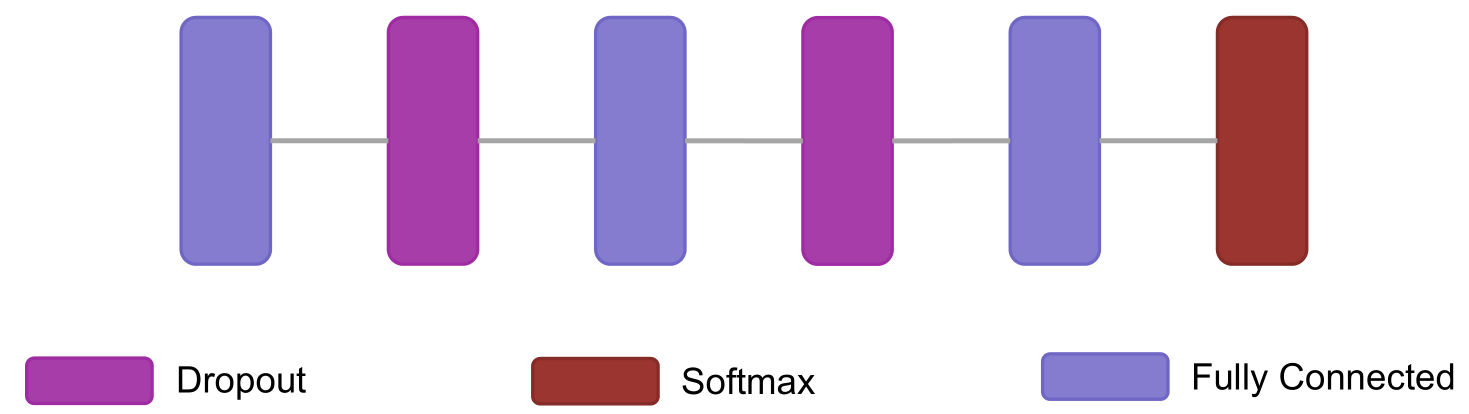
随着互联网技术的不断发展和应用，网络在给人们提供丰富资源和极大便利的同时，伴随而来的是互联网系统的安全性问题，验证码技术正是为了应对这些特殊情况而诞生。验证码能有效防止某个黑客对某一个特定注册用户用特定程序暴力破解方式进行不断的登陆尝试，不少网站为了防止用户利用机器人自动注册、登录、灌水，都采用了验证码技术。验证码为互联网安全作出了很大贡献，但也存在漏洞，为了更好地防止恶意攻击、保护网站安全，研究验证码自动识别算法，能够及早发现验证码设计存在的漏洞，并据此作出调整以增强验证码的安全性。

一方面，随着网络安全技术及验证码生成技术的不断发展，验证码生成技术变得越来越复杂，验证码的种类也越来越多。另一方面，当前的验证码识别系统具有很强的针对性，不同的验证码需要不同的图像预处理、分割和识别算法，对于给定的验证码生成系统，在获知其特点后，通过一定的识别策略往往能够以一定的准确率进行识别。

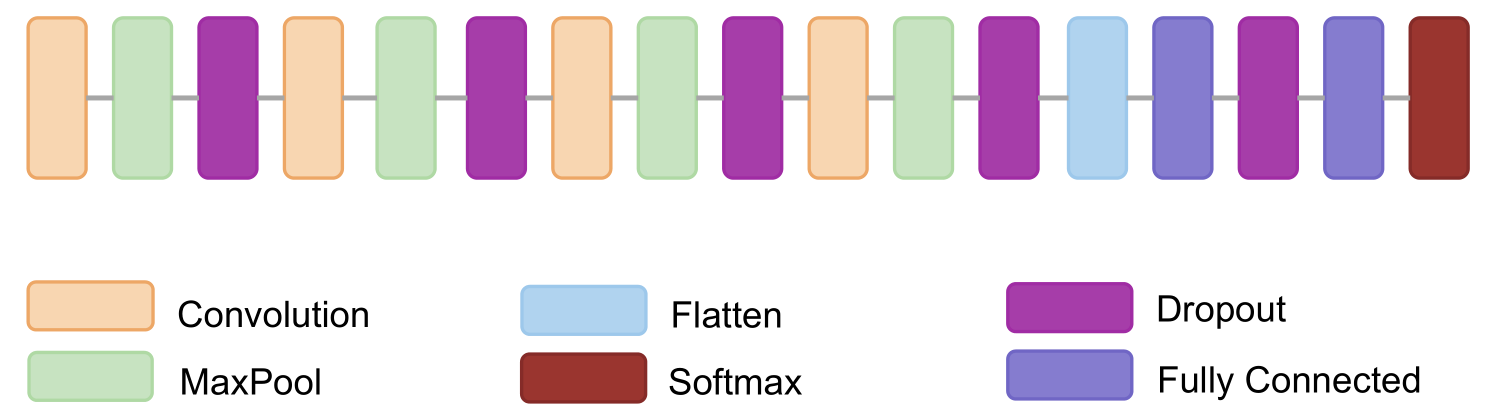
该模块共进行了四种验证码的识别，分别为简单纯数字/字母，混合数字字母，变形四则运算四种形式，输出为各类验证码图片上的验证结果。

### 6.2.2模型结构

1.数字四则运算，有噪点干扰，输出计算结果



2. 英文字母+数字验证码，包含 5 个字符，有噪点干扰，文字无旋转形变，验证方法为要求用户输出验证码中的字符，大小写不限（ 为验证方便可统一转为大写 ）。



3英文字母+数字验证码，包含 4 个字符，有噪点干扰，文字有旋转形变，验证方法为要求用户输出验证码中的字符。

