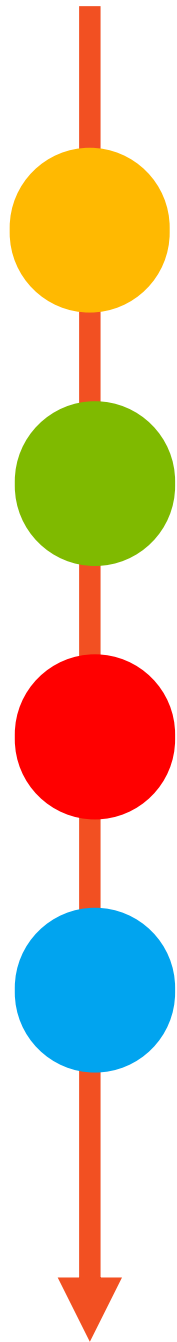


2018 TJMSC Tech. Courses

# Machine Learning Practice

Dinghow Yang  
Tongji Microsoft Student Club

Nov 4, 2018  
Room 516, Ji Shi Building  
SSE, Tongji Univ



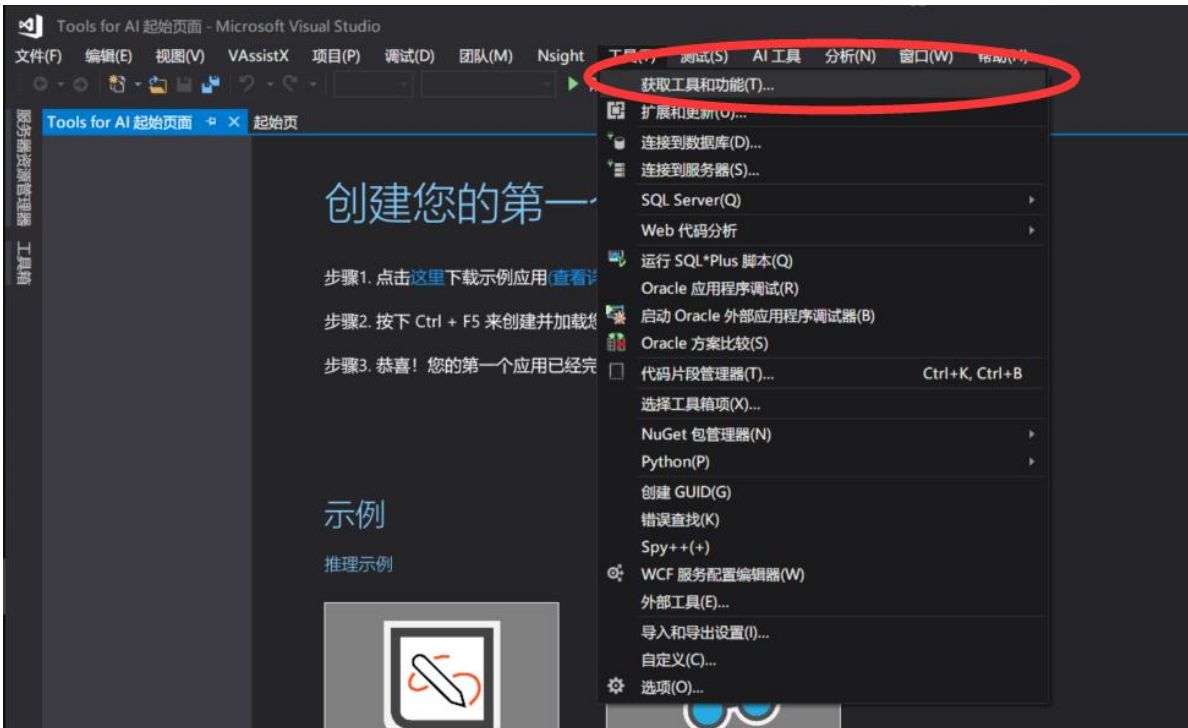
Visual Studio & Tools for AI Configuration

Train model by Custom Vision

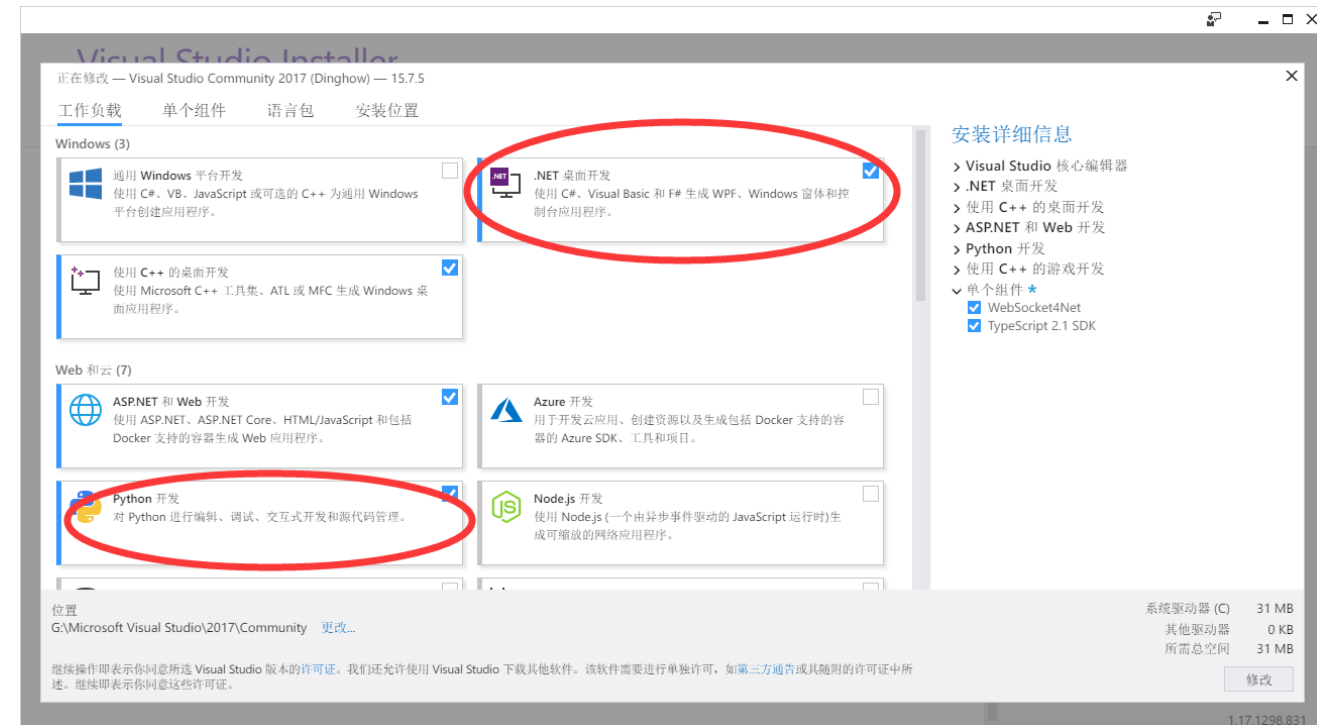
Realize a Window Form Application

Introduction of other Azure AI services

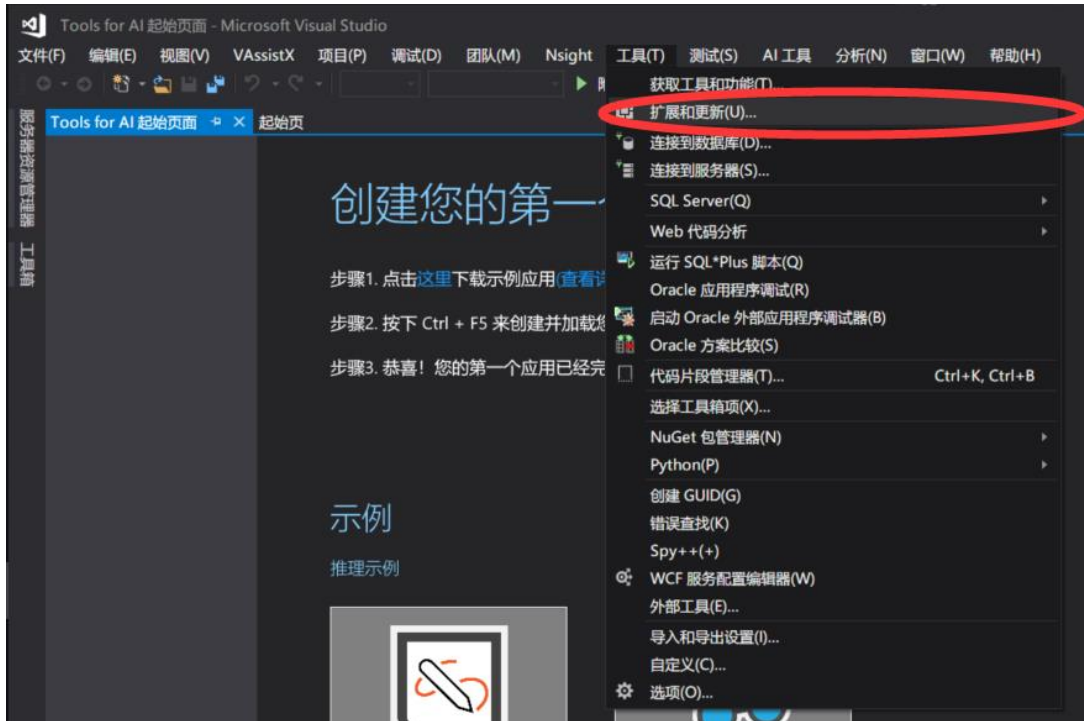
# Visual Studio Configuration



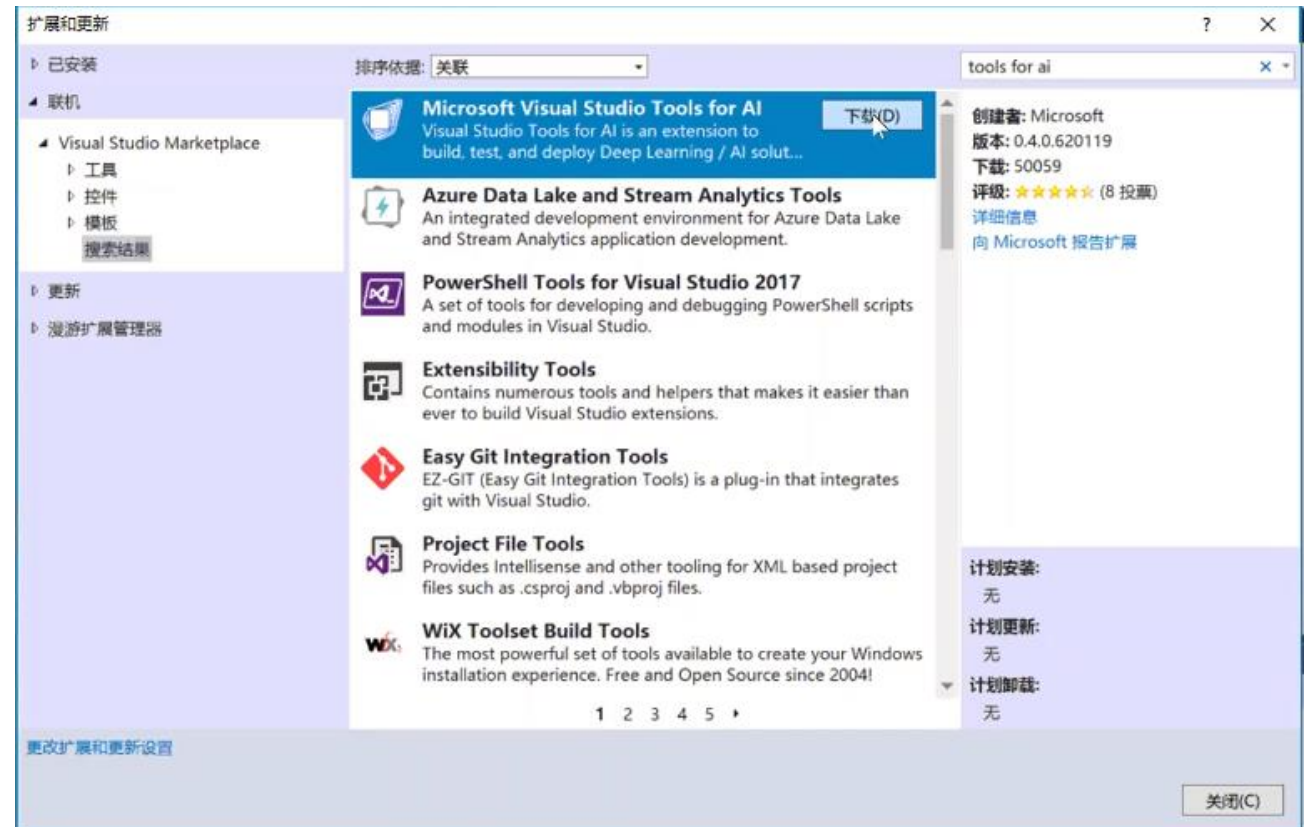
- .NET桌面开发
- Python开发



# Tools for AI Configuration



## Install Tools for AI



# Train model by Custom Vision

---

## What is Custom Vision?

The Azure Custom Vision API is a cognitive service that lets you build, deploy and improve custom image classifiers. An image classifier is an AI service that sorts images into classes (tags) according to certain characteristics. Unlike the Computer Vision service, Custom Vision allows you to create your own classifications.

<https://customvision.ai>

# Train model by Custom Vision

- Create new project
- Upload & tag images

Create new project

Name\*

Test

Description

Enter project description

Project Types ⓘ

☒ Classification

☐ Object Detection (preview)

Classification Types ⓘ

☒ Multilabel (Multiple tags per image)

☐ Multiclass (Single tag per image)

Domains ⓘ

☐ General

☐ Food

☐ Landmarks

☐ Retail

☐ Adult

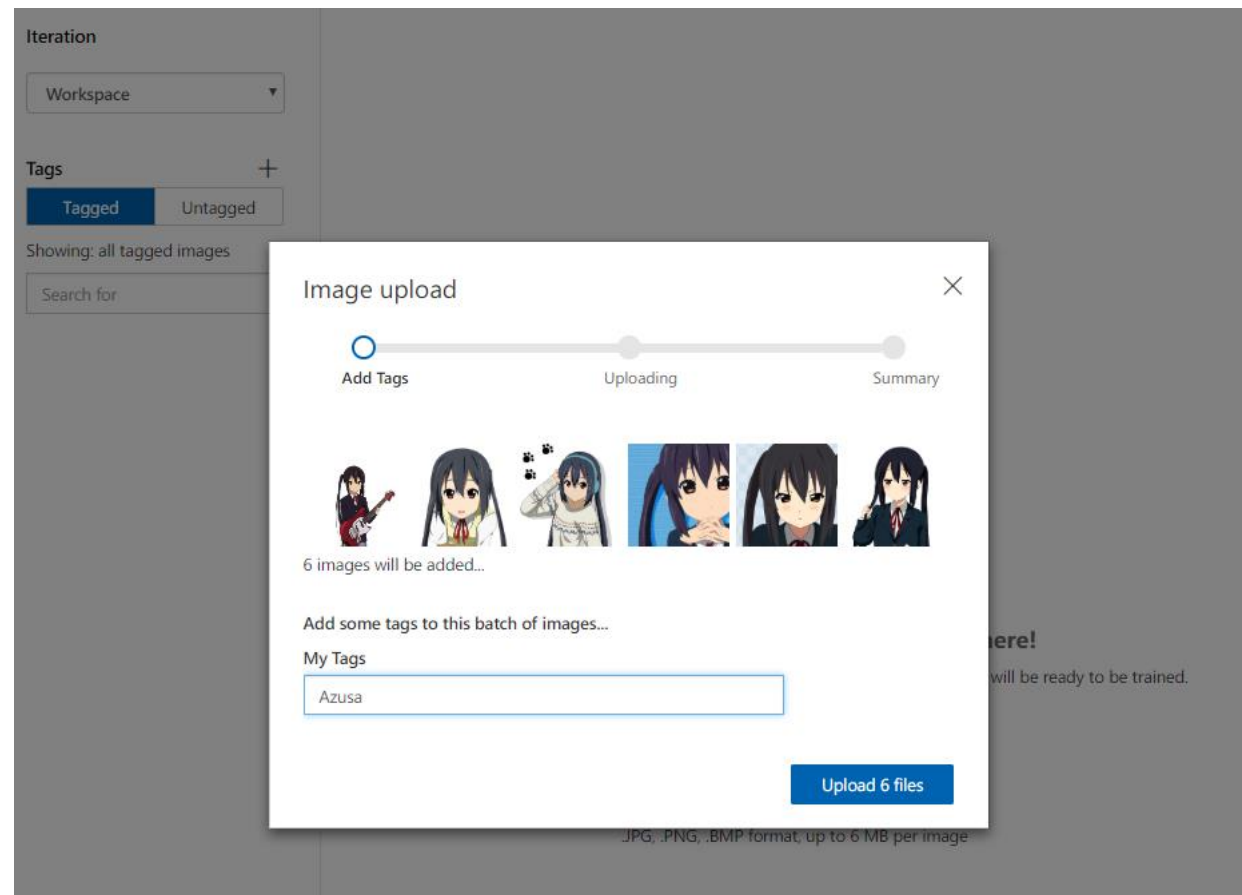
☒ General (compact)

☐ Landmarks (compact)

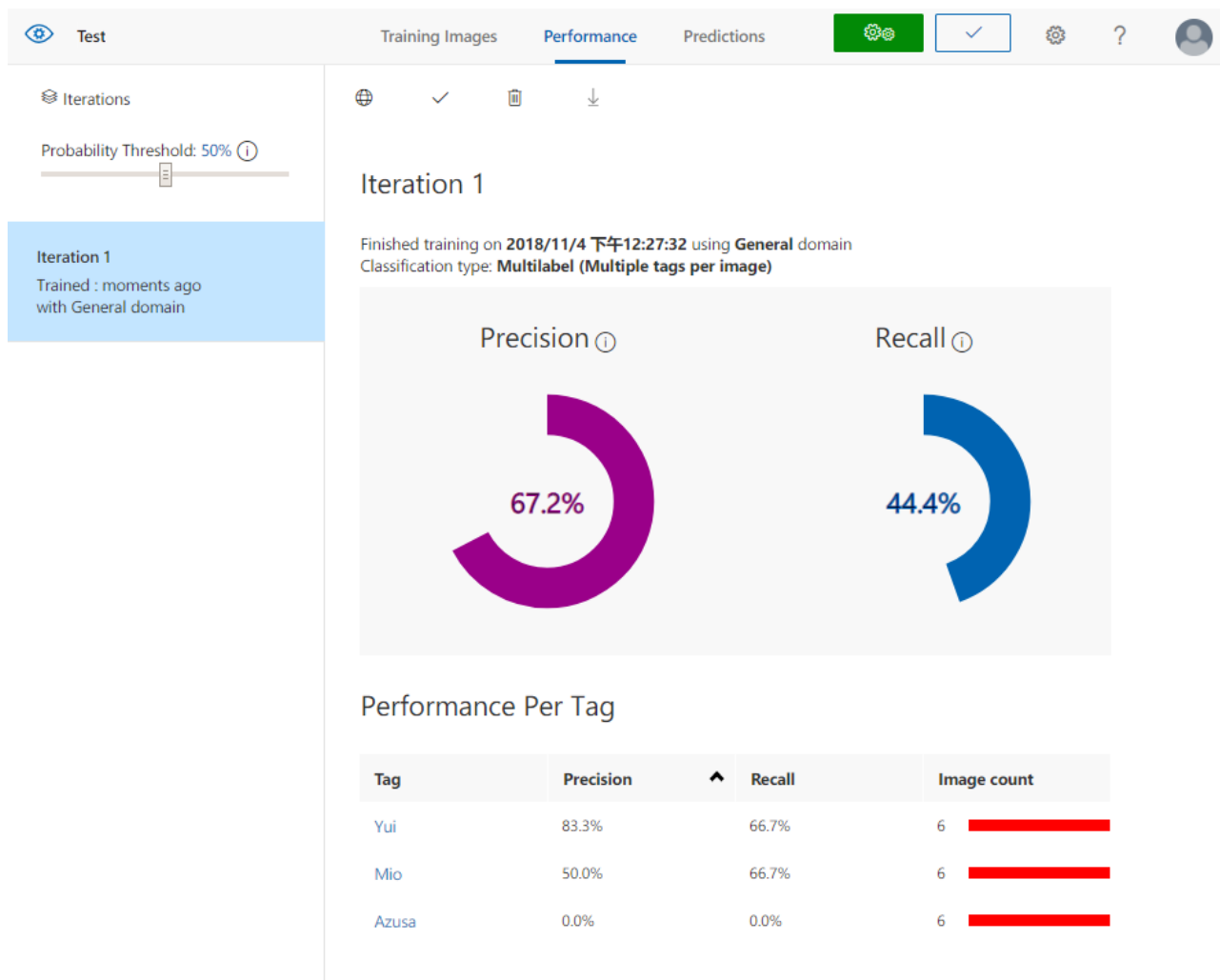
☐ Retail (compact)

Cancel

Create project

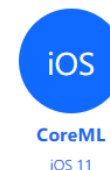


# Train model by Custom Vision

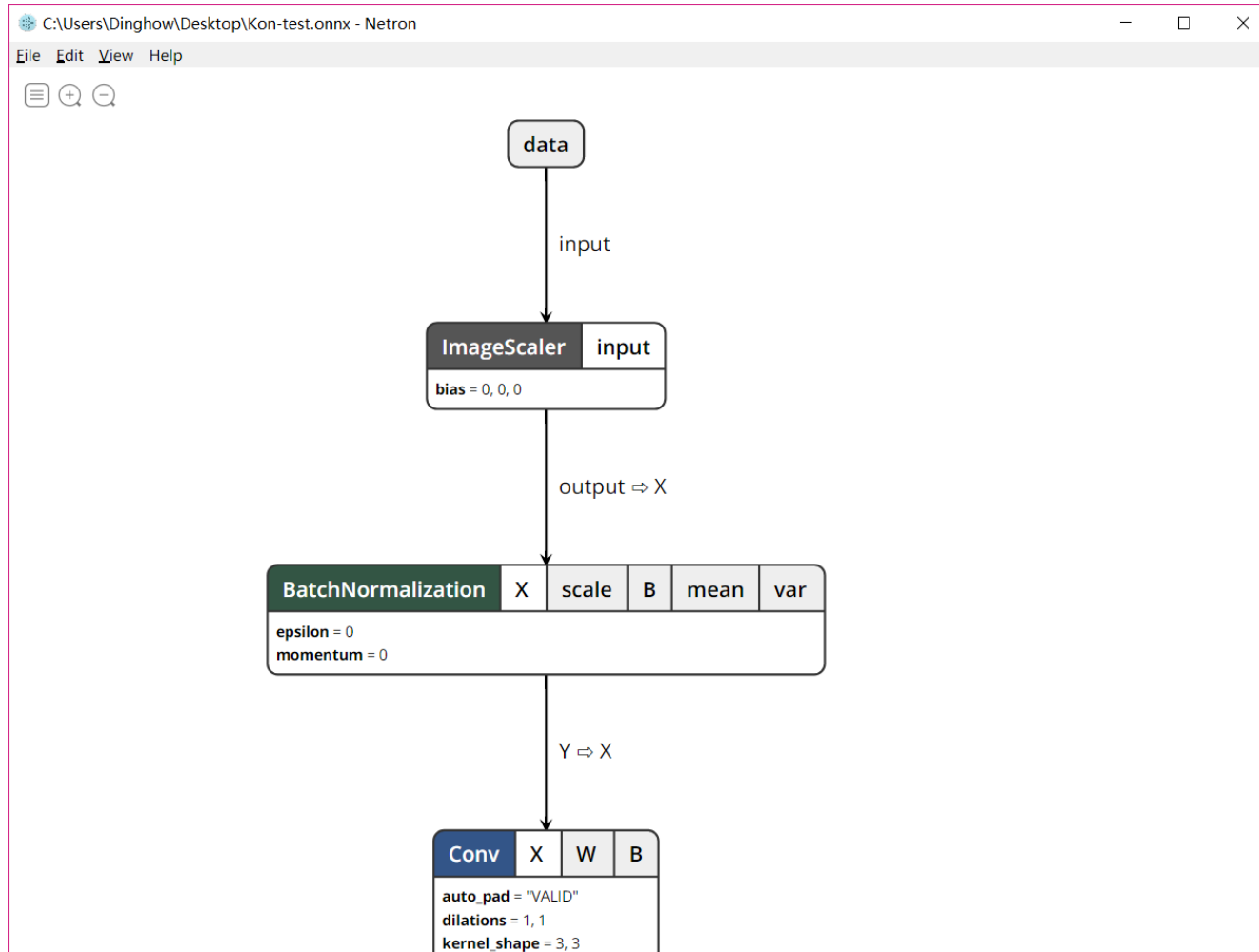


- Train model
- Download your model

Choose your platform



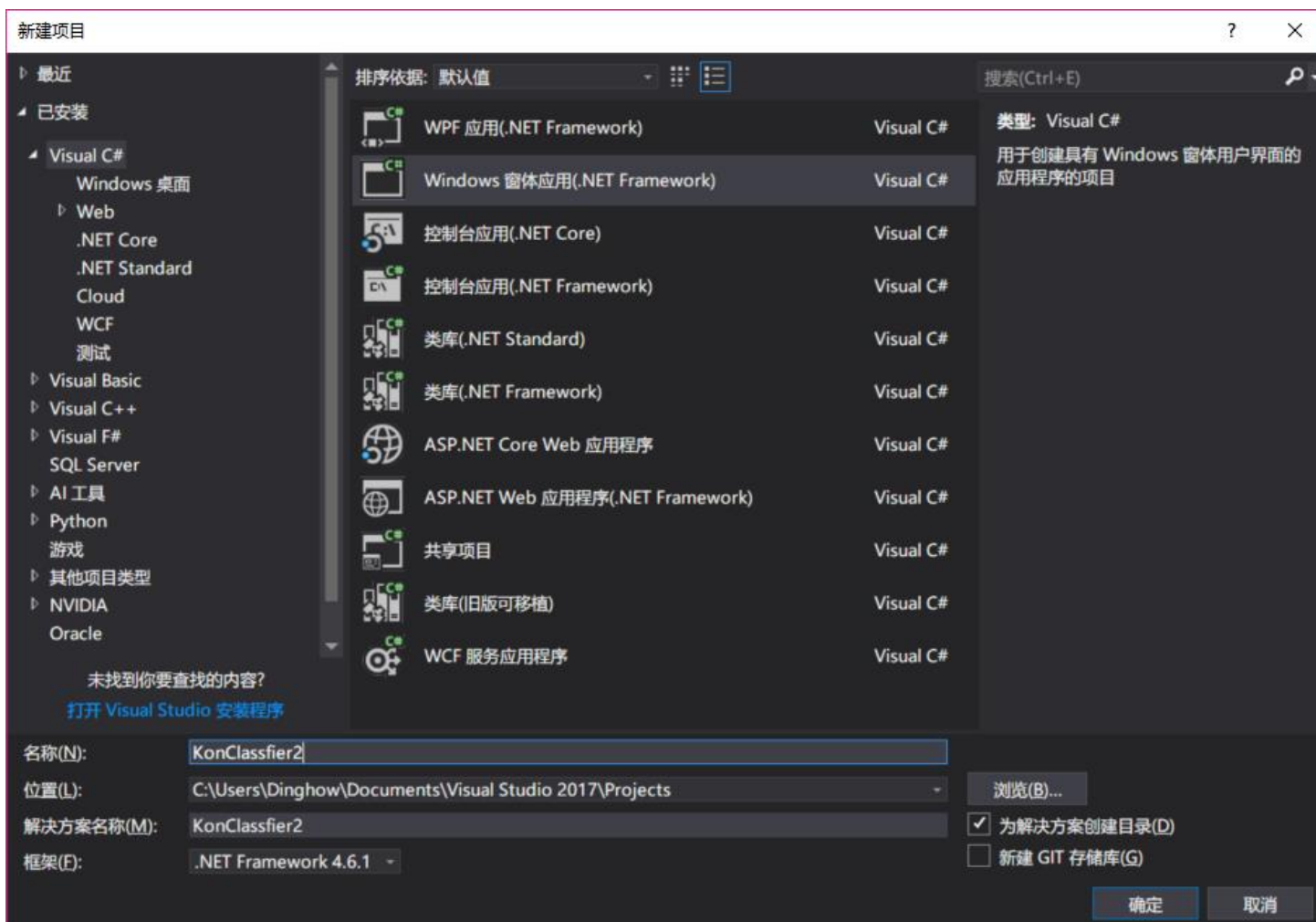
# Train model by Custom Vision



- View and Edit model by Netron



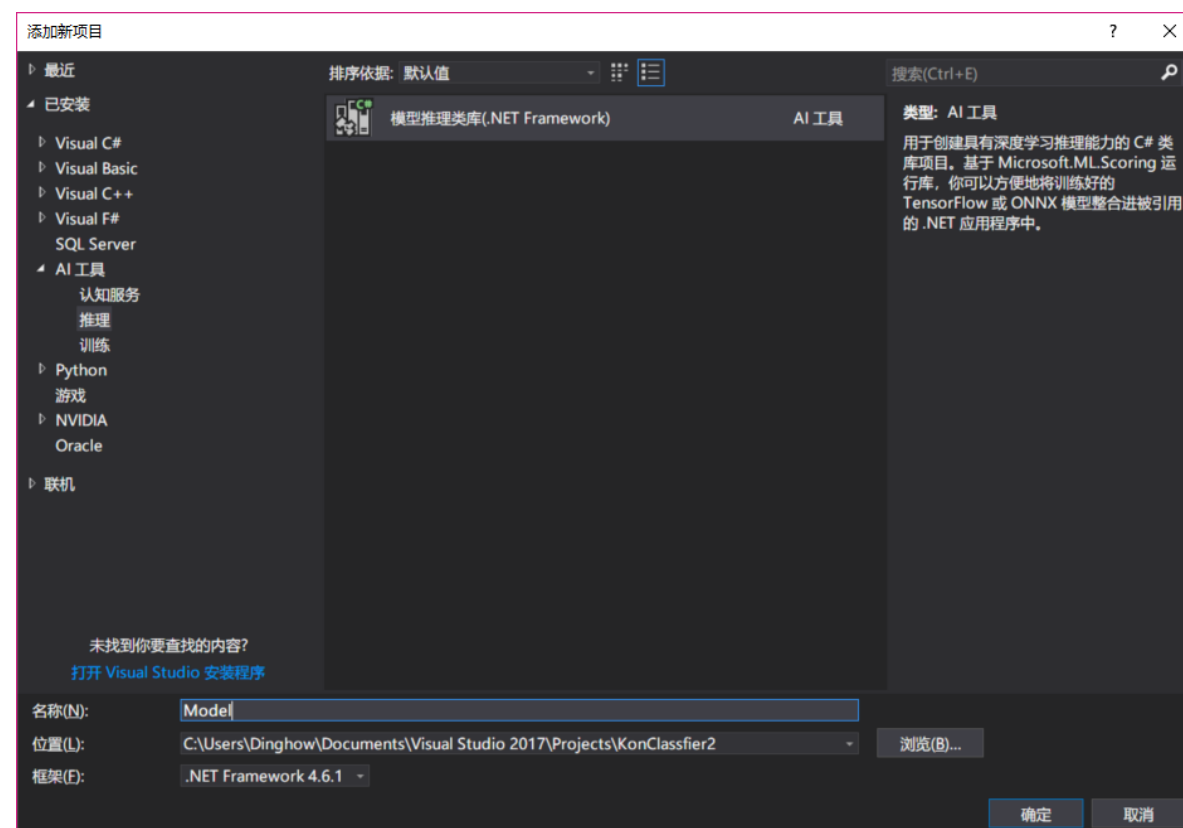
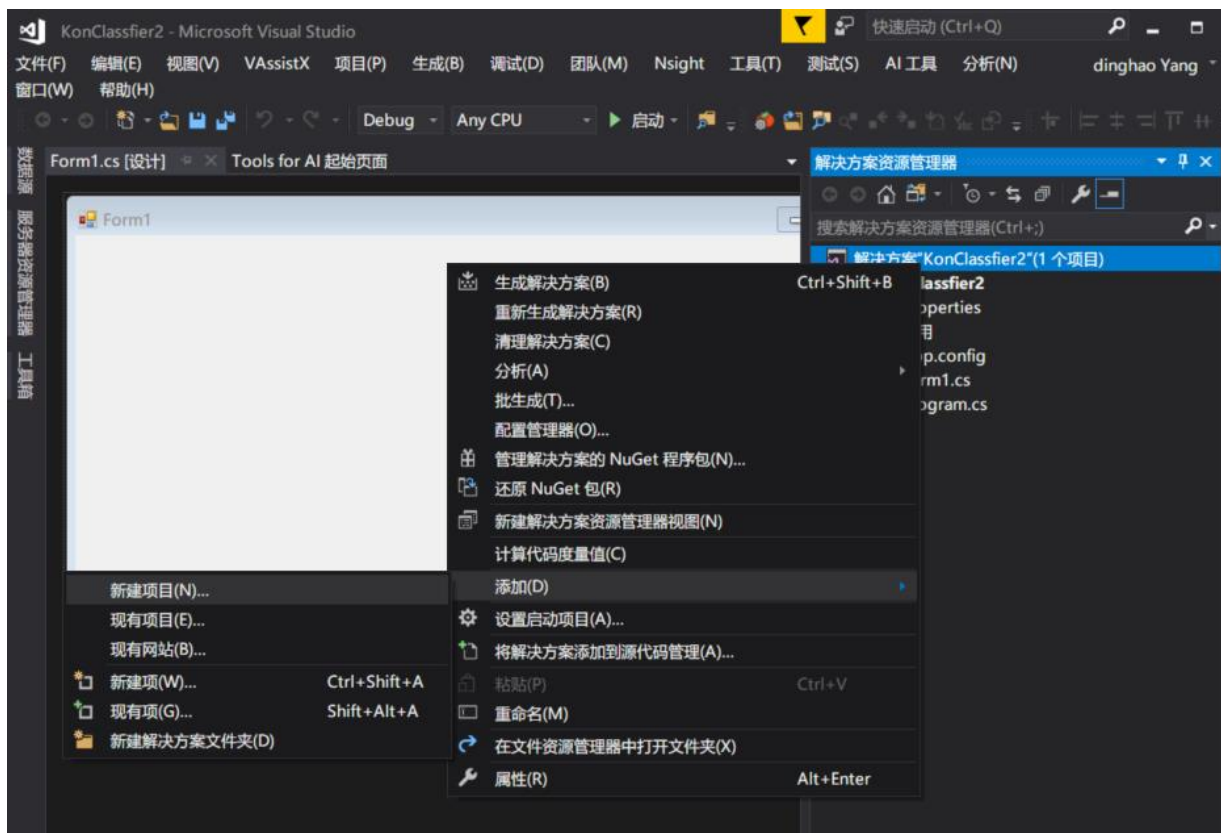
# Realize a Window Form Application



- Create a WFA project

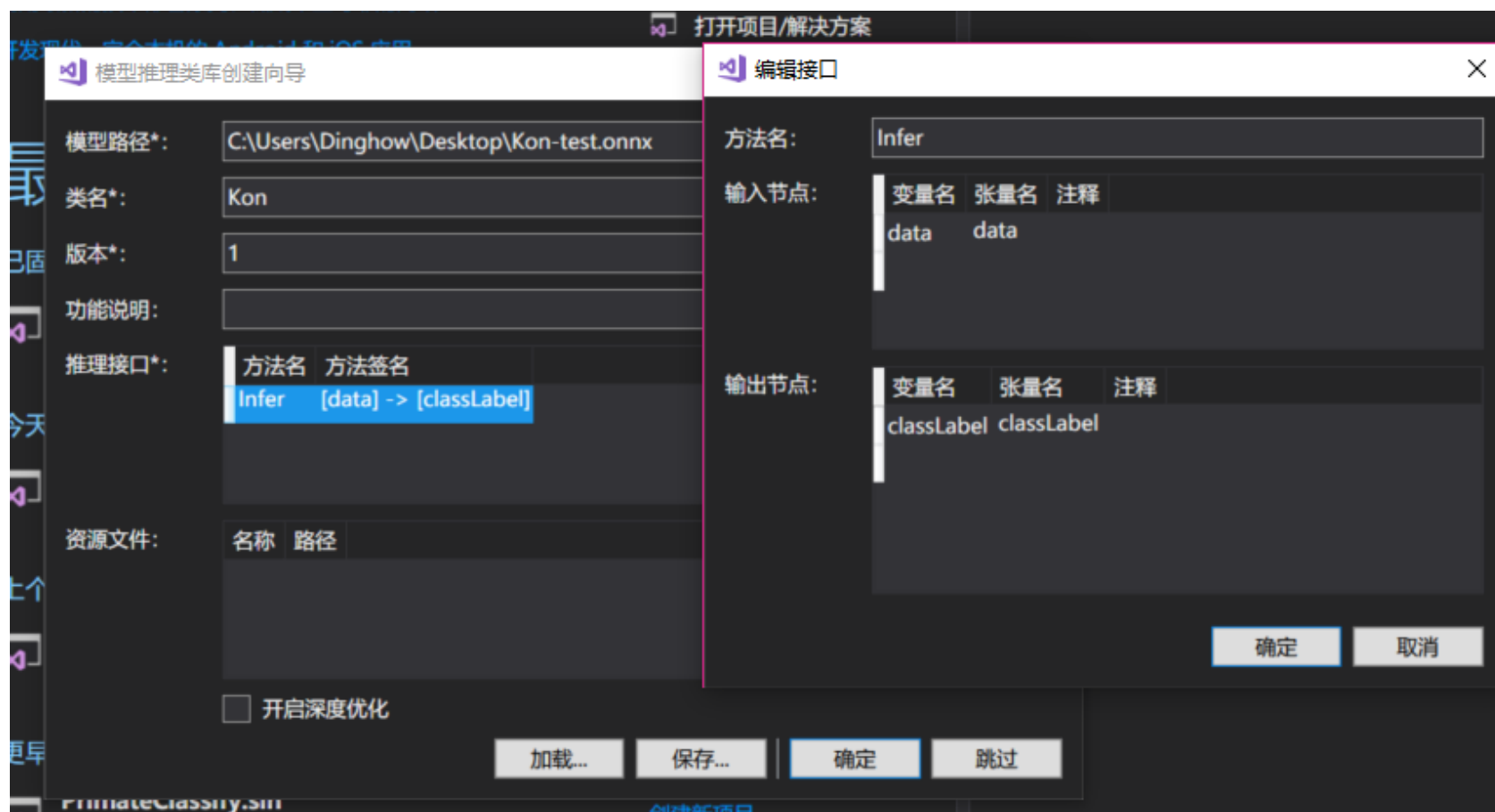
# Realize a Window Form Application

- Add an AI Model project



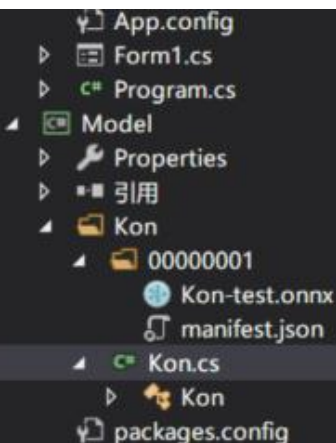
# Realize a Window Form Application

- Configure your model



# Realize a Window Form Application

```
45 /// </summary>
46 public IEnumerable<IEnumerable<string>> Infer(IEnumerable<IEnum
47 {
48     List<float> dataCombined = new List<float>();
49     foreach (var input in dataBatch)
50     {
51         dataCombined.AddRange(input);
52     }
53
54     List<Tensor> result = manager.RunModel(
55         modelName,
56         int.MaxValue,
57         inferInputNames,
58         new List<Tensor> { new Tensor(dataCombined, new List<lc
59         inferOutputNames
60     );
61
62     List<string> r0 = new List<string>();
63     result[0].CopyTo(r0);
64     return r0;
65 }
```

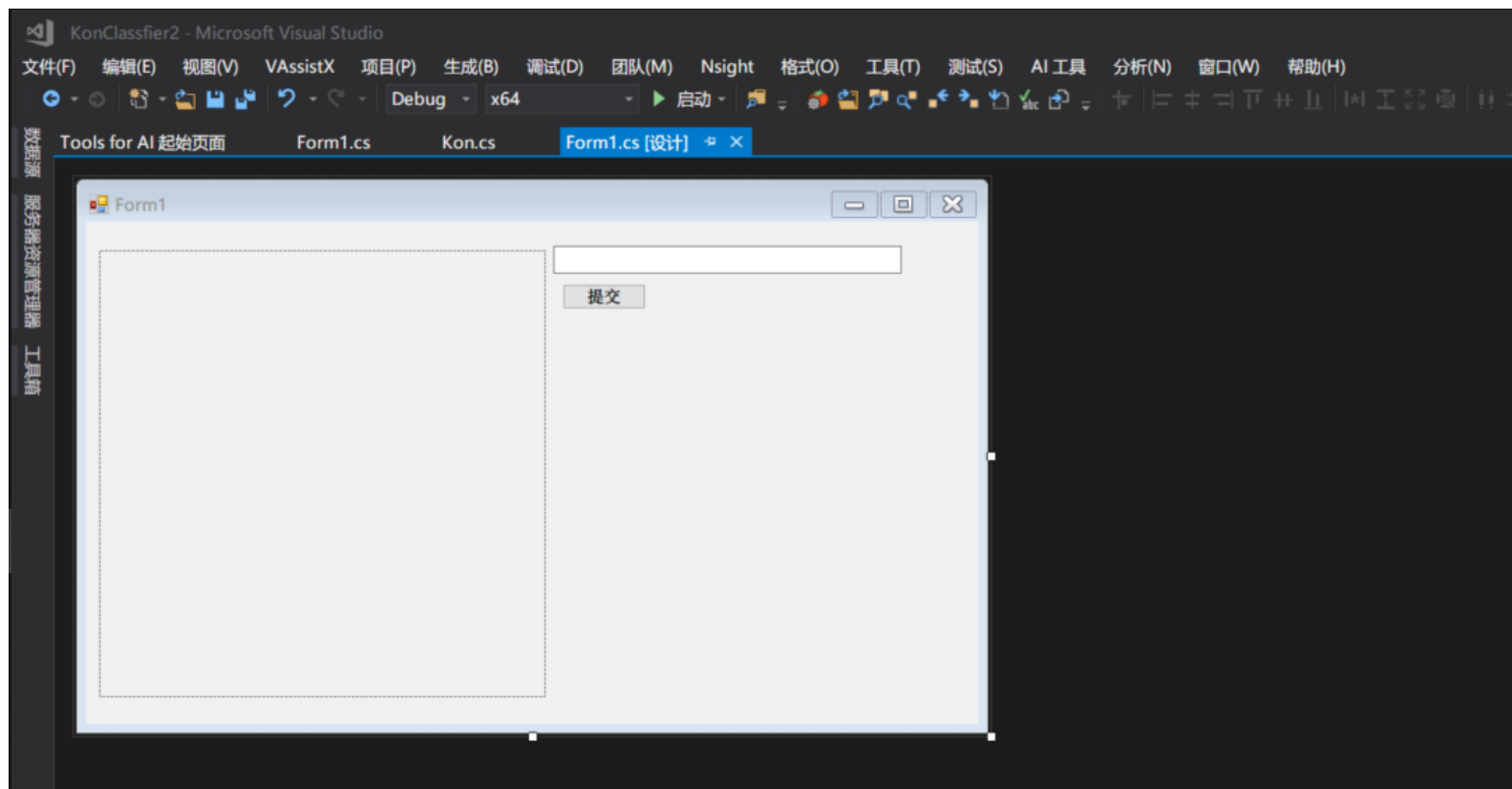


- Edit codes in model class



```
56         int.MaxValue,
57         inferInputNames,
58         new List<Tensor> { new Tensor(dataCombined, new List<lc
59         inferOutputNames
60     );
61
62     List<string> r0 = new List<string>();
63     result[0].CopyTo(r0);
64
65     List<List<string>> results = new List<List<string>>();
66     results.Add(r0);
67
68     return results;
69 }
```

# Realize a Window Form Application



- Add elements to your window

# Realize a Window Form Application

```
11 namespace KonClassifier2
12 {
13     public partial class Form1 : Form
14     {
15         public Form1()
16         {
17             InitializeComponent();
18         }
19
20         private const int imageSize = 227;
21         private Model.Kon model;
22
23         private void Form1_Load(object sender, EventArgs e)
24         {
25             model = new Model.Kon();
26         }
27
28         private void button1_Click(object sender, EventArgs e)
29         {
30             bool isSuccess = false;
31
32             label1.Text = string.Empty;
33             pictureBox1.Image = null;
34             pictureBox1.Refresh();
35
36             try
37             {
38                 pictureBox1.Load(textBox1.Text);
39                 isSuccess = true;
40             }
41             catch (Exception ex)
42             {
43                 MessageBox.Show("图片读取错误!", ex.Message);
44             }
45
46             if (isSuccess)
47             {
48                 //Reform the input image
```

```
35
36
37         try
38         {
39             pictureBox1.Load(textBox1.Text);
40             isSuccess = true;
41         }
42         catch (Exception ex)
43         {
44             MessageBox.Show("图片读取错误!", ex.Message);
45         }
46
47         if (isSuccess)
48         {
49             //Reform the input image
50             Bitmap clonedBmp = new Bitmap(imageSize, imageSize);
51             Graphics gNormalized = Graphics.FromImage(clonedBmp);
52
53             gNormalized.DrawImage(pictureBox1.Image, 0, 0, imageSize, imageSize);
54
55             var imageArray = new float[imageSize * imageSize * 3];
56             for (int y = 0; y < imageSize; y++)
57             {
58                 for (int x = 0; x < imageSize; x++)
59                 {
60                     var color = clonedBmp.GetPixel(x, y);
61
62                     imageArray[y * imageSize + x] = color.B;
63                     imageArray[y * imageSize + x + imageSize * imageSize] = color.G;
64                     imageArray[y * imageSize + x + 2 * imageSize * imageSize] = color.R;
65                 }
66             }
67
68             //Get result by accessing model
69             var result = model.Infer(new List<IEnumerable<float>> { imageArray }).First().First();
70
71             label1.Text = result;
72         }
73     }
74 }
```

- Add functions



# Realize a Window Form Application

```
11 namespace KonClassifier2
12 {
13     public partial class Form1 : Form
14     {
15         public Form1()
16         {
17             InitializeComponent();
18         }
19
20         private const int imageSize = 227;
21         private Model.Kon model;
22
23         private void Form1_Load(object sender, EventArgs e)
24         {
25             model = new Model.Kon();
26         }
27
28         private void button1_Click(object sender, EventArgs e)
29         {
30             bool isSuccess = false;
31
32             label1.Text = string.Empty;
33             pictureBox1.Image = null;
34             pictureBox1.Refresh();
35
36             try
37             {
38                 pictureBox1.Load(textBox1.Text);
39                 isSuccess = true;
40             }
41             catch (Exception ex)
42             {
43                 MessageBox.Show("图片读取错误!", ex.Message);
44             }
45
46             if (isSuccess)
47             {
48                 //Reform the input image
```

```
35
36         {
37             try
38             {
39                 pictureBox1.Load(textBox1.Text);
40                 isSuccess = true;
41             }
42             catch (Exception ex)
43             {
44                 MessageBox.Show("图片读取错误!", ex.Message);
45             }
46
47             if (isSuccess)
48             {
49                 //Reform the input image
50                 Bitmap clonedBmp = new Bitmap(imageSize, imageSize);
51                 Graphics gNormalized = Graphics.FromImage(clonedBmp);
52
53                 gNormalized.DrawImage(pictureBox1.Image, 0, 0, imageSize, imageSize);
54
55                 var imageArray = new float[imageSize * imageSize * 3];
56                 for (int y = 0; y < imageSize; y++)
57                 {
58                     for (int x = 0; x < imageSize; x++)
59                     {
60                         var color = clonedBmp.GetPixel(x, y);
61
62                         imageArray[y * imageSize + x] = color.B;
63                         imageArray[y * imageSize + x + imageSize * imageSize] = color.G;
64                         imageArray[y * imageSize + x + 2 * imageSize * imageSize] = color.R;
65                     }
66                 }
67
68                 //Get result by accessing model
69                 var result = model.Infer(new List<IEnumerable<float>> { imageArray }).First().First();
70
71                 label1.Text = result;
72             }
73         }
74     }
75 }
```

- Add functions

# Realize a Window Form Application



- Run your application!



# Introduction of other Azure AI services



## 使用 AI 解决业务问题



### 影像

图像处理算法能智能标识图像、描述图像 和调整图像大小。



### 知识

通过映射复杂信息和数据来解决任务，例如 智能建议和语义搜索。



### 语言

允许应用通过预生成脚本处理自然语言、评估情绪，并了解如何识别用户需求。



### 语音

将语言音频转换为文本，使用声音进行验证， 或向应用添加识别。



### 搜索

将必应搜索 API 添加到应用之中， 通过单个 API 调用梳理数  
页、图像、视频和新闻。

# Introduction of other Azure AI services

Microsoft  
认知服务

API 文档 > API 参考

**POST** Analyze Image

**POST** Describe Image

**POST** Get Thumbnail

**GET** List Domain Specific Models

**POST** OCR

**POST** Recognize Domain  
Specific Content

**POST** Tag Image

## Computer Vision API - v1.0

[API definition](#)

The Computer Vision API provides state-of-the-art algorithms to process images and return information. For example, it can be used to determine if an image contains mature content, or it can be used to find all the faces in an image. It also has other features like estimating dominant and accent colors, categorizing the content of images, and describing an image with complete English sentences. Additionally, it can also intelligently generate images thumbnails for displaying large images effectively.

### Analyze Image

This operation extracts a rich set of visual features based on the image content.

Two input methods are supported -- (1) Uploading an image or (2) specifying an image URL. Within your request, there is an optional parameter to allow you to choose which features to return. By default, image categories are returned in the response.

A successful response will be returned in JSON. If the request failed, the response will contain an error code and a message to help understand what went wrong.

Http Method

POST

[打开 API 测试控制台](#)

Request URL

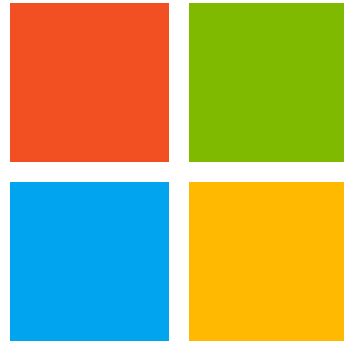
`https://api.cognitive.azure.cn/vision/v1.0/analyze[?visualFeatures][&details][&language]`

Request parameters

# Learning Resources

---

- 微软AI B站主页: <https://space.bilibili.com/333935914/>
- 微软ML环境搭建:  
<https://www.cnblogs.com/ms-uap/p/9123033.html>
- Custom Vision: <https://customvision.ai/>
- Custom Vision Demo制作:  
<https://www.bilibili.com/video/av26004062>
- Azure认知服务: <https://dev.cognitive.azure.cn/>



Microsoft



微软学生俱乐部

扫一扫二维码，加入该群。



关注TJMSC微信公众号