北京大学
PEKING UNIVERSITY

k210图像分类案例：Baidu Flower

李国玮

北京大学 计算机学院

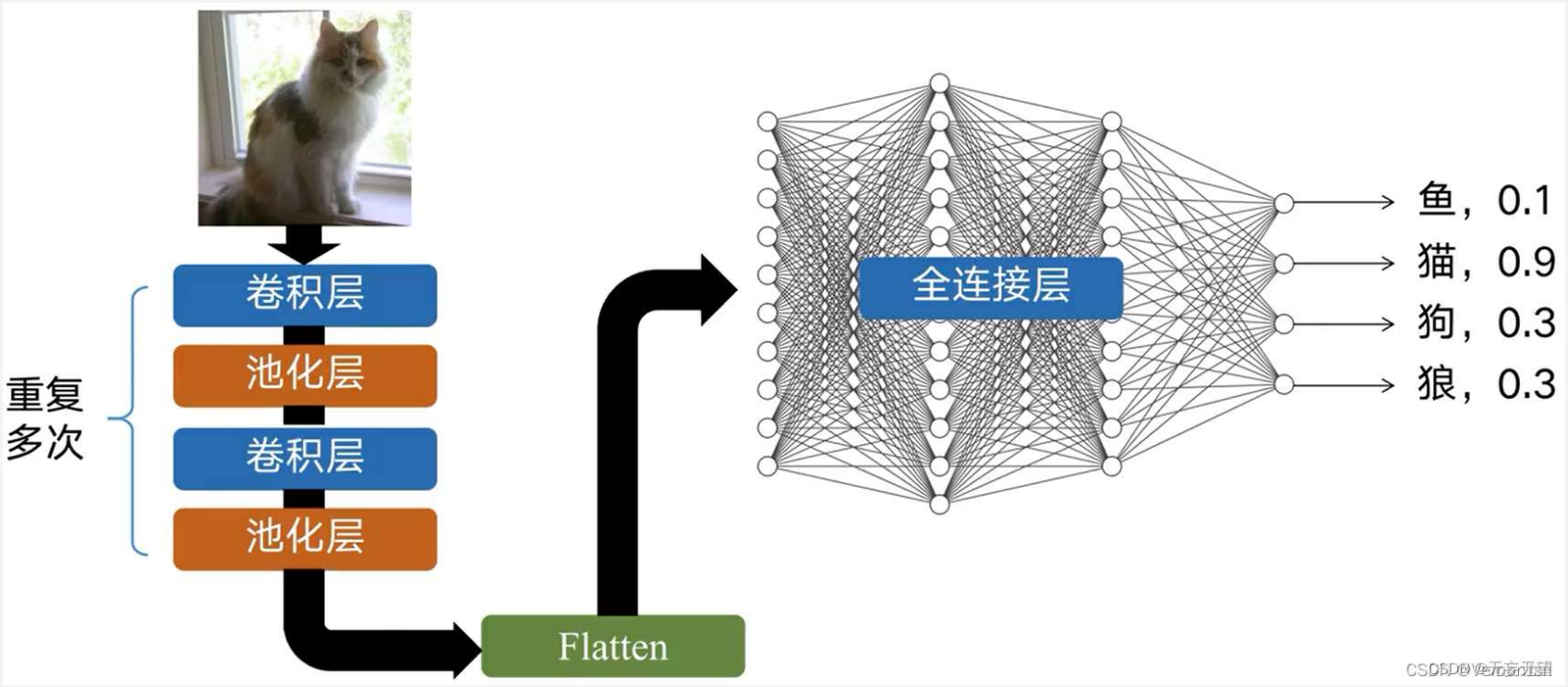
2024.12.04

01 图像分类问题

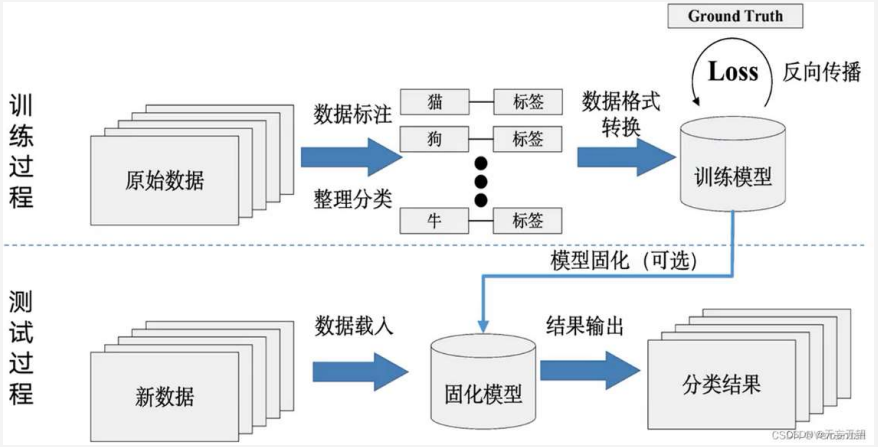
北京大学 PEKING UNIVERSITY

图像分类任务的目标：根据输入的图像，由神经网络模型输出图像所属的类别。

K210采用的神经网络模型为：卷积神经网络（Convolutional Neural Network，CNN）



02 训练模型



创建模型：

- 输入层：接收图像数据，格式为224x224x3（RGB三通道）
- 卷积层：通过滑动卷积核在输入图像上进行计算，提取图像特征
- 池化层：降低卷积层对位置的敏感性，防止过拟合
- 全连接层：对前面提取的特征进行整合，并通过激活函数输出每个类别的概率

加载模型参数：将预训练好的模型参数赋给创建好的模型，减少训练时间

模型训练：

- 前向传播：输入数据通过各个网络层的计算，直到模型输出层
- 反向传播：计算损失函数关于每个模型参数的梯度
- 优化器：根据反向传播得到的梯度信息调整模型的参数，目标降低损失函数

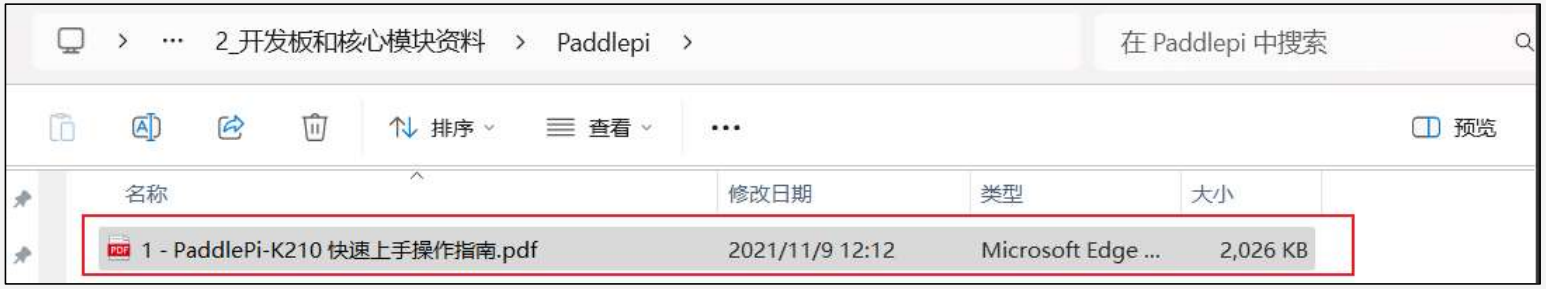
模型评估：固定模型参数，评估模型在测试数据集上的分类准确率

03 模型转换

普通的模型并不能很好地运行在开发板等特定硬件上，为了在特定硬件上部署， 需要借助一些工具，进行模型转换：

- **模型量化：**压缩模型，降低模型数值的精度（浮点数->整数）来减少模型的存储空间和计算开销，同时尽量保持模型的性能
- **模型转换：**通过nncase工具，把模型转换成K210可以运行的模型mobilenet.kmodel

以上详细内容可通过该文件找到：



04 例程讲解

北京大学 PEKING UNIVERSITY

main函数开头与camera_lcd例程相同，初始化中断控制器、电源、IO引脚、LCD屏幕、DVP摄像头

```
170  int main()
171  {
172      /* Set CPU and dvp clk */
173      sysctl_pll_set_freq(SYSCTL_PLL0, PLL0_OUTPUT_FREQ);
174      sysctl_pll_set_freq(SYSCTL_PLL1, PLL1_OUTPUT_FREQ);
175      sysctl_clock_enable(SYSCTL_CLOCK_AI);
176      // uarths_init();
177      plic_init();
178      io_set_power();
179      io_init();
180
181      /* LCD init */
182      printf("LCD init\n");
183      lcd_init();
184      lcd_set_direction(DIR_YX_RLDU);
185      lcd_clear(BLACK);
186      /* DVP init */
187      printf("DVP init\n");
188      dvp_init(8);
189      dvp_set_xclk_rate(24000000);
190      dvp_enable_burst();
```

- 允许DVP将图像输出用于AI和Display

```
dvp_set_output_enable(0, 1);
dvp_set_output_enable(1, 1);
dvp_set_image_format(DVP_CFG_RGB_FORMAT);
dvp_set_image_size(320, 240);
```

04 例程讲解

北京大学 PEKING UNIVERSITY

```
49   static image_t kpu_image, display_image, crop_image;
```

```
typedef struct
{
    uint8_t *addr;
    uint16_t width;
    uint16_t height;
    uint16_t pixel;
    uint16_t format;
} image_t;
```

```
kpu_image.pixel = 3;
kpu_image.width = 320;
kpu_image.height = 240;
image_init(&kpu_image);
display_image.pixel = 2;
display_image.width = 320;
display_image.height = 240;
image_init(&display_image);
crop_image.pixel = 3;
crop_image.width = 224;
crop_image.height = 224;
image_init(&crop_image);
```

3个表示图像数据的全局变量

- kpu_image：DVP直接为AI传入的图像
- display_image：DVP直接为显示传入的图像
- crop_image：用于KPU进行图像分类的图像

image_t各字段含义：

- addr：图像的内存地址（由image_init函数分配）
- width：图像宽度
- height:：图像高度
- pixel：一个像素的字节数

04 例程讲解

设置用于AI和显示的内存地址，DVP会将每帧图像同时传输到这两片内存区域，其中：

- AI：像素的R、G、B值各占1字节，所有像素的R值依次保存在kpu_image.addr，G值在kpu_image.addr+width*height, B值在kpu_image.addr+2*width*height
- 显示：像素的R、G、B值各占5, 6, 5 bits，共2字节。所有像素依次保存在display_image.addr

```
dvp_set_ai_addr((uint32_t)kpu_image.addr, (uint32_t)(kpu_image.addr + 320 * 240),
                (uint32_t)(kpu_image.addr + 320 * 240 * 2));
dvp_set_display_addr((uint32_t)display_image.addr);
```

04 例程讲解

北京大学 PEKING UNIVERSITY

```
#define INCBIN_STYLE INCBIN_STYLE_SNAKE
#define INCBIN_PREFIX
#include "incbin.h"
```

```
kpu_model_context_t task;

INCBIN(model, "mobilenet.kmodel");
```

```
/* init model */
if (kpu_load_kmodel(&task, model_data) != 0)
{
    printf("Cannot load kmodel.\n");
    return(-1);
}
```

- INCBIN宏：

将当前目录下的mobilenet.kmodel文件，加载到内存，并以3个全局变量存在：

const unsigned char model_data[];

const unsigned char *model_end;

const unsigned int model_size;

通过INCBIN，程序会将已训练好的模型加载到model_data

- kpu_load_kmodel

将模型信息model_data保存在task变量

04 例程讲解

北京大学 PEKING UNIVERSITY

- DVP向kpu_image和display_image传送帧

```
    while (1)
    {
#if (BOARD_VERSION == BOARD_V1_3)
        if (KEY_PRESS == key_get())
        {
            camera_switch();
        }
#endif

        g_dvp_finish_flag = 0;
        dvp_clear_interrupt(DVP_STS_FRAME_START | DVP_STS_FRAME_FINISH);
        dvp_config_interrupt(DVP_CFG_START_INT_ENABLE | DVP_CFG_FINISH_INT_ENABLE, 1);
        while (g_dvp_finish_flag == 0)
            ;
```

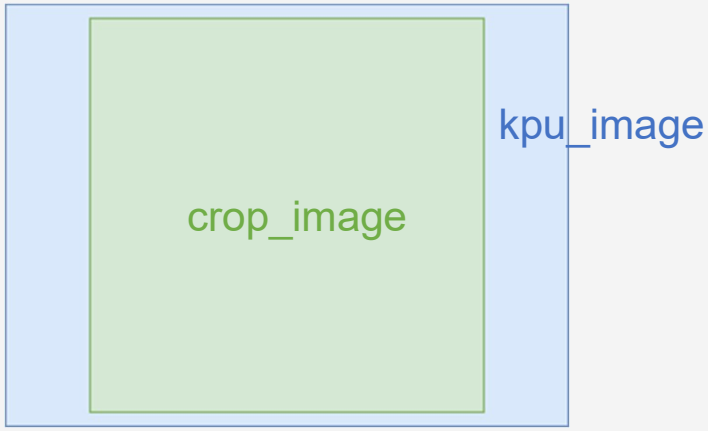
```
// 图像裁剪，讲kpu_image中(48, 8)为左上角的图像，裁剪到crop_image中
image_crop(&kpu_image, &crop_image, 48, 8);

g_ai_done_flag = 0;

if (kpu_run_kmodel(&task, crop_image.addr, 5, ai_done, NULL) != 0)
{
    printf("Cannot run kmodel.\n");
    return(-1);
}
while (!g_ai_done_flag);
```

- 将kpu_image(320*240)的中心区域裁剪到crop_image(224*224)
- **kpu_run_kmodel**：

运行task表示的模型，模型输入是crop_image，DMA通道为5，运行结束后调用ai_done函数



04 例程讲解

北京大学 PEKING UNIVERSITY

```c
static int ai_done(void* userdata)
{
    g_ai_done_flag = 1;
    float *features;
    size_t count;
    kpu_get_output(&task, 0, (uint8_t **)&features, &count);
    count /= sizeof(float);

    size_t i;
    for (i = 0; i < count; i++)
    {
        if (i % 64 == 0)
            printf("\n");
        printf("%f, ", features[i]);
    }

    printf("\n");
    return 0;
}
```

- ai_done：模型运行结束后的回调函数
- kpu_get_output：

将模型第0个输出层的输出结果，保存在features数组中，

输出结果是一个浮点数数组，元素个数为count，

每个元素表示图片属于某类的概率

04 例程讲解

北京大学 PEKING UNIVERSITY

```c
int main()
    while (1)

        float *features;
        size_t output_size;
        kpu_get_output(&task, 0, &features, &output_size);
        size_t cls = argmax(features, 5);

        const char *text = NULL;
        switch (cls)
        {
            case 0:
                text = "daisy";
                break;
            case 1:
                text = "dandelion";
                break;
            case 2:
                text = "roses";
                break;
            case 3:
                text = "sunflowers";
                break;
            case 4:
                text = "tulip";
                break;
        }

        /* display pic*/
        if (features[cls] > PROB_THRESH)
            ram_draw_string(display_image.addr, 150, 20, text, RED);
        lcd_draw_picture(0, 0, 320, 240, (uint32_t *)display_image.addr);
    }
```

- 在主函数中，通过argmax函数选择概率值最大对应的类别，即 最大值 所在的数组索引
- 根据索引，输出花的类别
- 如果概率大于0.7，则在LCD上显示该图片及图片的类别

THANK YOU