1 量化前后对比

量化前卷积操作耗时如图 1，图 2 所示，量化后卷积操作耗时如图 3，图 4 所示。

```
In [6]: # 执行第五层卷积层IP核
start = time.clock()
RunConv(conv_ip,3,3,1,1,1,0,conv5_in,weights,bias,conv5_output)
end = time.clock()
print(end-start)

104.50768000000001
```

图 1 量化前第 5 层卷积操作耗时

```
In [10]: # 执行第九层全连接层IP核，conv9 output需要去量化
start = time.clock()
RunConv(conv_ip,1,1,1,1,1,0,conv9_in,weights,bias,output)
end = time.clock()
print(end-start)

11.712597999999986
```

图 2 量化前第 9 层卷积操作耗时

```
In [32]: # 执行第五层卷积层IP核，conv5_output需要去量化
start = time.clock()
RunConv(conv_ip,3,3,1,1,1,0,conv5_in,weights,bias,conv5_output,maxw5_w)
end = time.clock()
print(end-start)

88.51469199999997
```

图 3 量化后第 5 层卷积操作耗时

```
In [43]: # 执行第九层全连接层IP核，conv9 output需要去量化
start = time.clock()
RunConv(conv_ip,1,1,1,1,1,0,conv9_in,weights,bias,output,maxw9_w)
end = time.clock()
print(end-start)

10.091155999999955
```

图 4 量化后第 9 层卷积操作耗时

量化前输出图片如图 5 所示，量化后输出图片如图 6 所示。通过对比图片可以看出，量化后识别出的矩形框仍相同，但识别的置信度有所降低。

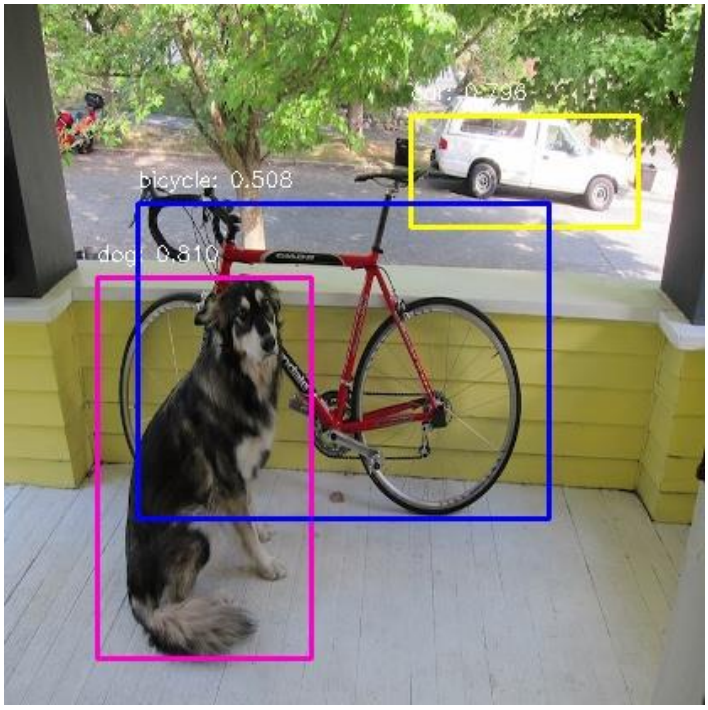

图 5 量化前输出图片

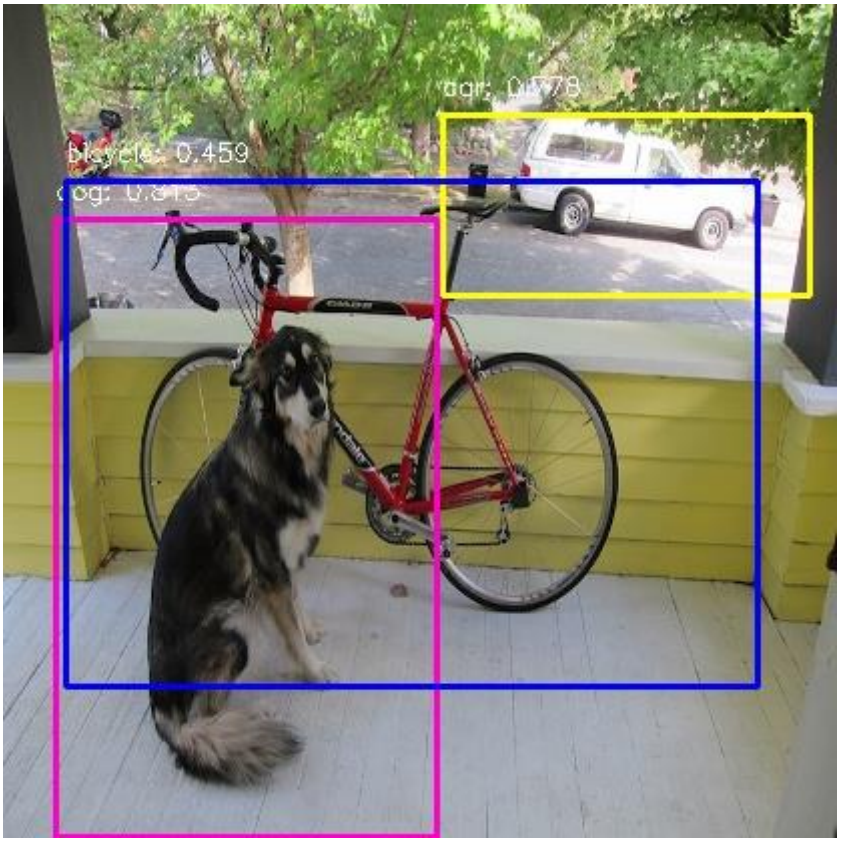

图 6 量化后输出图片

2. 优化前后对比

优化前卷积 ip 核的分析报告如图 7，优化后的分析报告如图 8.

Performance Estimates

Timing (ns)

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	10.00	8.750	1.25

Latency (clock cycles)

Summary

Latency		Interval		
min	max	min	max	Type
2173720	76616818712	2173720	76616818712	none

Detail

Instance

Loop

	Latency			Initiation Interval			
Loop Name	min	max	Iteration Latency	achieved	target	Trip Count	Pipelined
- channel	2173696	76616818688	8491 ~ 74821112	-	-	256 ~ 1024	no
+ ffeature_row	8489	74821110	653 ~ 2877735	-	-	13 ~ 26	no
++ feature_col	650	2877732	50 ~ 110682	-	-	13 ~ 26	no
+++ weight_row	5	110637	5 ~ 36879	-	-	1 ~ 3	no
++++ weight_col	2	36876	2 ~ 12292	-	-	1 ~ 3	no
+++++ Input_Channel	1536	12288	12	-	-	128 ~ 1024	no

Utilization Estimates

Summary

Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	5	-	-
Expression	-	6	0	1613
FIFO	-	-	-	-
Instance	2	5	2889	3926
Memory	-	-	-	-
Multiplexer	-	-	-	668
Register	-	-	2147	-
Total	2	16	5036	6207
Available	120	80	35200	17600
Utilization (%)	1	20	14	35

图 7 优化后分析报告

Performance Estimates

Timing (ns)

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	10.00	9.483	1.25

Latency (clock cycles)

Summary

Latency		Interval		
min	max	min	max	Type
2336283	12898902043	2336283	12898902043	none

Detail

Instance

Loop

	Latency			Initiation Interval			
Loop Name	min	max	Iteration Latency	achieved	target	Trip Count	Pipelined
- channel_ffeature_row_feature_col	2336257	12898902017	54 ~ 18634	-	-	43264 ~ 692224	no
+ weight_row_weight_col	5	18585	5 ~ 2065	-	-	1 ~ 9	no
++ Input_Channel	266	2058	12	2	2	128 ~ 1024	yes

Utilization Estimates

Summary

Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	12	-	-
Expression	-	10	0	1961
FIFO	-	-	-	-
Instance	2	5	2889	3926
Memory	-	-	-	-
Multiplexer	-	-	-	644
Register	0	-	2688	96
Total	2	27	5577	6627
Available	120	80	35200	17600
Utilization (%)	1	33	15	37

图 8 优化后分析报告

通过对比可以看出，虽然优化后时钟频率有所下降，但由于并行性提高性能反而提升，同时资源消耗也有所增加。

优化后的运行时间如图 9，图 10 所示。

```
In [9]: # 执行第五层卷积层IP核，conv5_output需要去量化
        start = time.clock()
        RunConv(conv_ip,3,3,1,1,1,0,conv5_in,weights,bias,conv5_output,maxw5_w)
        end = time.clock()
        print(end-start)

        23.41226900000001
```

图 9 优化后第 5 层卷积操作耗时

```
In [16]: # 执行第九层全连接层IP核，conv9 output需要去量化
         start = time.clock()
         RunConv(conv_ip,1,1,1,1,1,0,conv9_in,weights,bias,output,maxw9_w)
         end = time.clock()
         print(end-start)

         2.605002000000013
```

图 10 优化后第 9 层卷积操作耗时

优化后输出图片如图 11 所示。可以看到比起优化前的输出，优化后的置信度进一步发生下降。

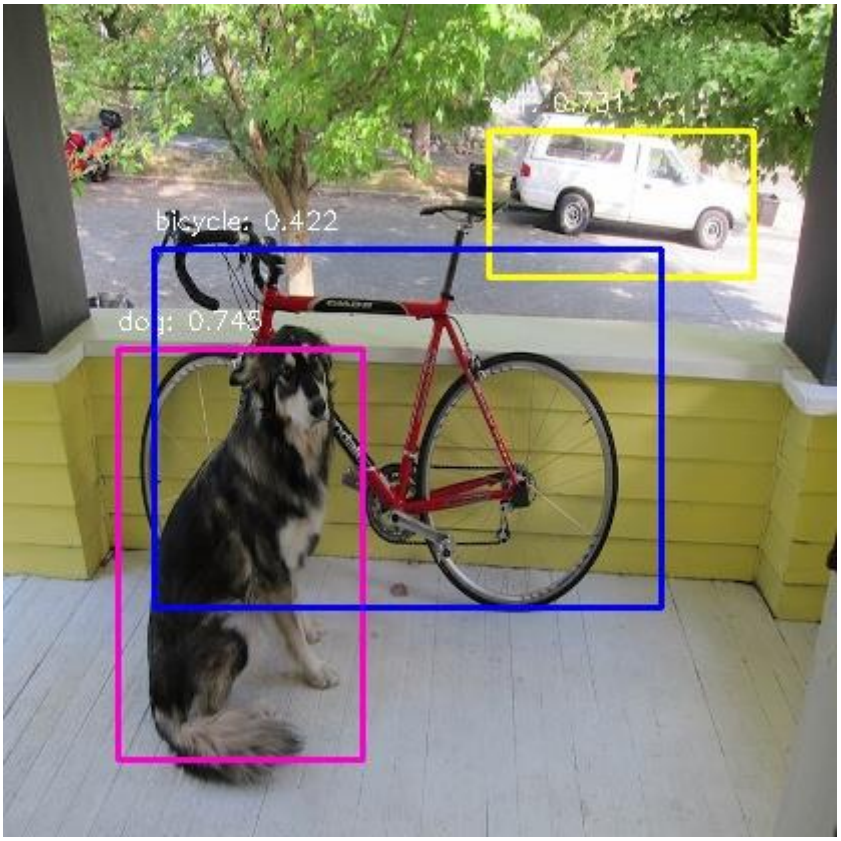

图 11 优化后输出图片