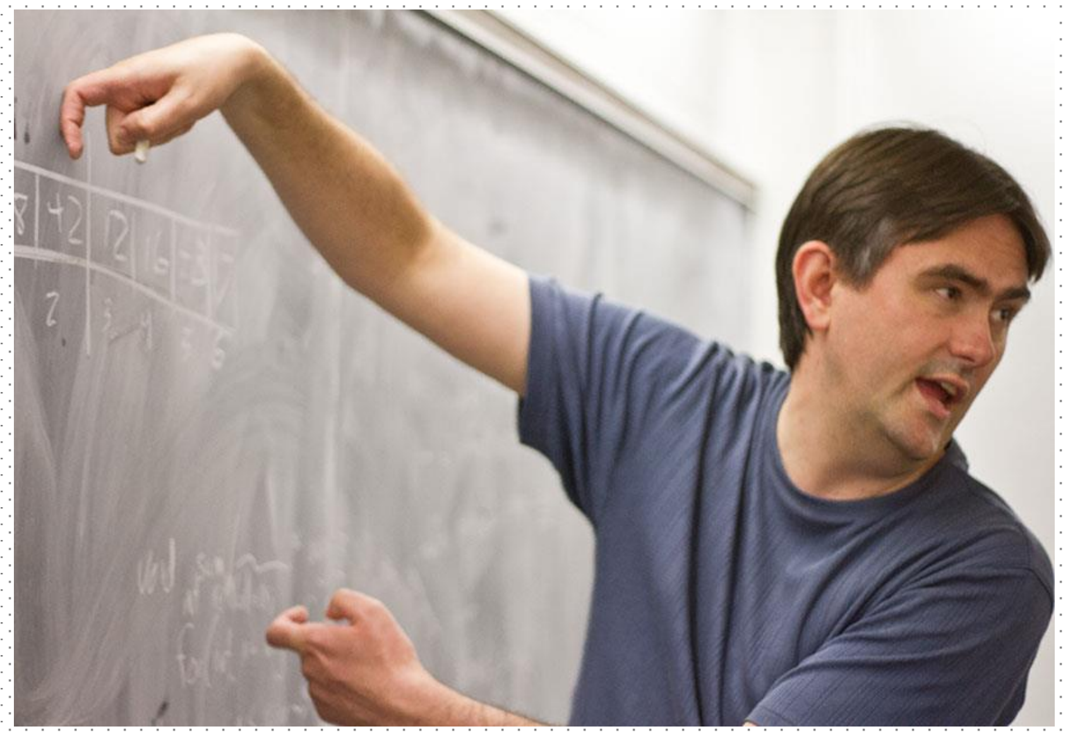
Facial Recognition Assignment on the Raspberry Pi

Arsalan Bin Najeeb
abnajeeb@knox.edu

Shebaz Chowdhury
fchowdhury@knox.edu

Joint work with:
Annie Song
afsong@knox.edu

Supervised by:
David Bunde
dbunde@knox.edu



What is Heterogeneous Computing?

Heterogeneous computing is the idea of using different types of processors together to solve complex problems. With heterogeneity, processes can be faster and more efficient, setting the foundation for future technologies to be run smoothly. There are examples of these types of systems around us everywhere; both smartphones and supercomputers typically use heterogeneous processors now.

Why Raspberry Pi?

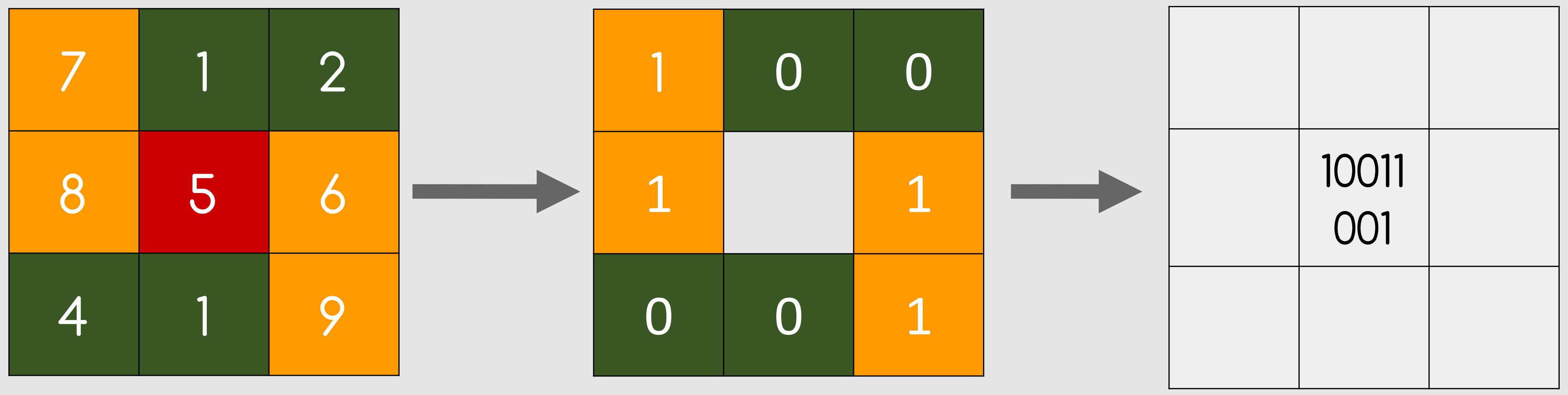
The Raspberry Pi is a small, low cost single-board computer used by hobbyists and educators. It comes in many different configurations and various peripherals can be used with it. It is also a Heterogeneous computing system with both a CPU and a dedicated GPU. Our configuration allows students to code in C and use the camera peripheral on the Pi.

Local Binary Patterns (LBP)

We calculate the average of Red Green and Blue values, getting a value between 0 to 255. We then form a binary number using Local Binary Patterns.

From each pixel, we compare with neighboring values. Neighbors that are smaller contribute to a 0 while larger neighbors contribute a 1. These values give the binary representation of that pixel's value.

Once we get our number, we repeat LBP for each non-bordering pixel. Then, we form a histogram for each image. We will compare each histogram to find a relation.



7	1	2
8	5	6
4	1	9

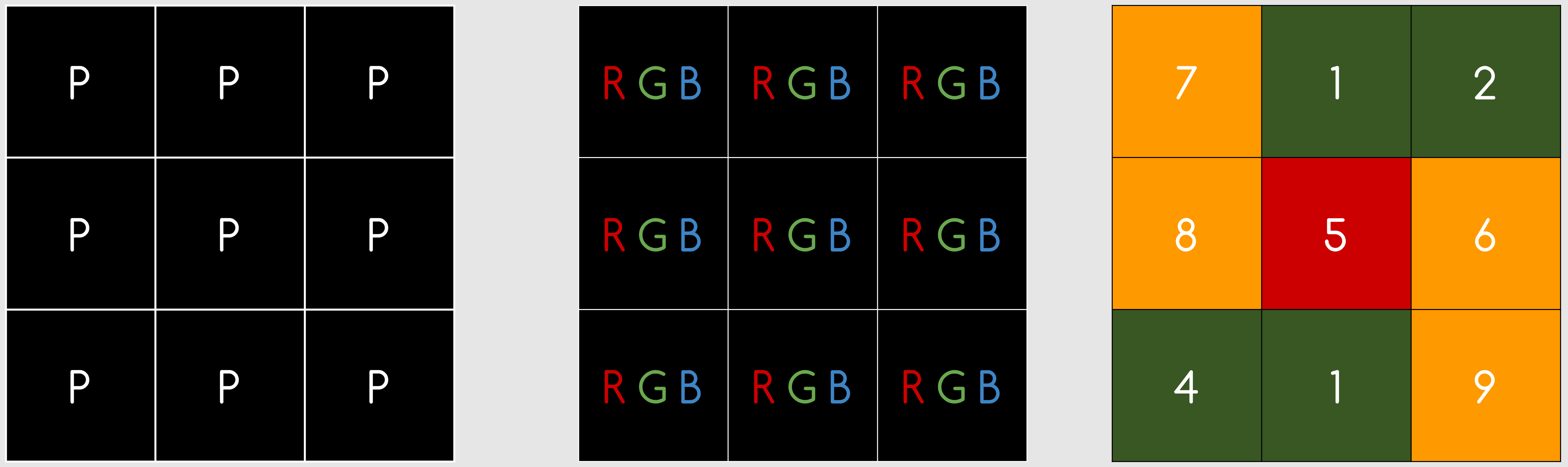
1	0	0
1		1
0	0	1

	10011 001	

Face Recognition Assignment

Each student will be provided with a Raspberry Pi with its camera pre-attached. We will also provide them with starter code in C that can take a picture and save it as a BMP file, which can be used to view the image that was taken. They will then write the algorithm that will use our implementation of LBP to break down their image into its pixels and further into its RGB values.

The follow-up assignment will have them parallelize their algorithm, compare running times, and calculate their speedup.



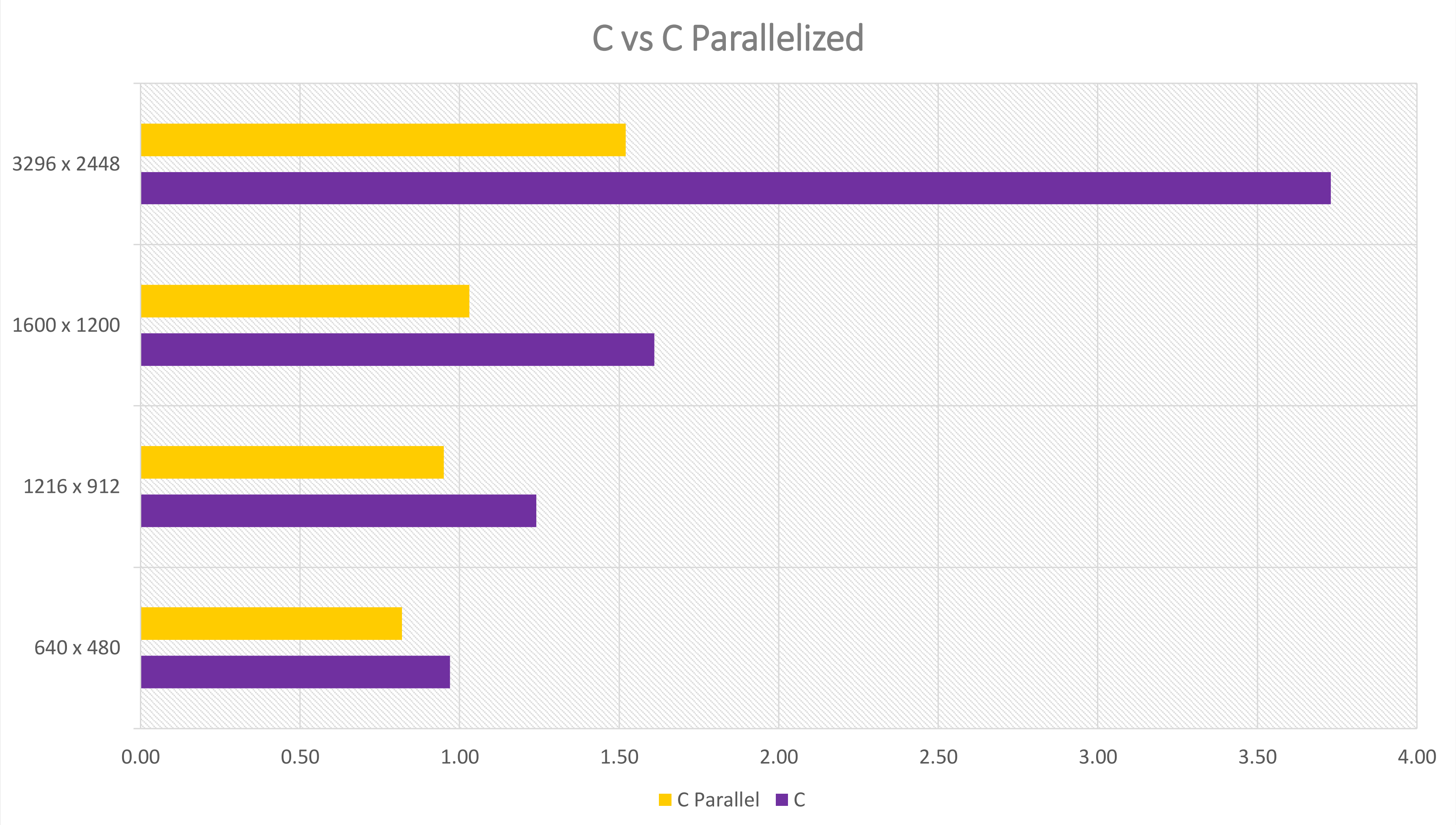
P	P	P
P	P	P
P	P	P

RGB	RGB	RGB
RGB	RGB	RGB
RGB	RGB	RGB

7	1	2
8	5	6
4	1	9

Value	0	1	2	...	255
Frequency	5000	100	215	...	50

Expected Run Time



This work was partially supported by the National Science Foundation under grant OAC-1829554.

The project was also partially supported by the Paul K & Evalyn Elizabeth Cook Richter Memorial Fund