

UNIVERSIDADE FEDERAL DE SANTA CATARINA

Metodologia Bayesiana para controle de qualidade de pizzas

**Florianópolis – SC
2008/2**

UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
CURSO DE CIÊNCIAS DA COMPUTAÇÃO

Metodologia Bayesiana para controle de qualidade de pizzas

Daniel Besen de Aguiar

Trabalho de Conclusão de curso apresentado como parte dos requisitos para obtenção do grau em Bacharel em Ciências da Computação

Florianópolis – SC
2008/2

Agradecimentos

Agradeço à todos aqueles que estiveram comigo nessa caminhada. Agradeço à minha família, principalmente aos meus pais por todo o apoio. Agradeço aos meus amigos pela motivação e suporte. Agradeço à minha Chefe Prof^a Márcia Mantelli pela oportunidade que me foi dada no Labucal. Agradeço ao meu orientador Sylvio Luiz Mantelli Neto por ter-me orientado durante a construção desse trabalho. Agradeço a toda equipe do LABTUCAL por ter me ajudado nesse processo, em especial Bianca Sens dos Santos, por ter-me sugerido a idéia da metodologia, por ter me fornecido material de pesquisa e auxiliado durante a execução. Agradeço também ao professor Aldo Von Wangenheim, por ter permitido a utilização de ferramentas e metodologias utilizadas em seu laboratório. Agradeço ao Elvis Presley pelos seus conselhos musicais e à banda AC/DC por sempre me deixar mais animado, mesmo nas tarefas mais enfadonhas.

Dedicatória

Dedico esse trabalho a todos aqueles que me ajudaram a tornar esse trabalho uma realidade. Meu orientador Sylvio Luiz Mantelli Neto. Agradeço à Bianca Sens dos Santos. Dedico esse trabalho especialmente para meus pais, irmão e amigos.

Sumário

1	Resumo.....	8
2	Introdução	9
3	Procedimento Utilizado	11
3.1	Aquisição das imagens	11
3.2	Criação das Máscaras	13
3.3	Extração dos Valores dos Pixels.....	18
3.4	Análise dos valores e determinação do intervalo.....	19
3.5	Análise sensorial	25
4	Conclusão	30
5	Referências	30
6	Anexos	32

Lista de Figuras

Figura 1: Curva de temperatura durante o processo de cocção.....	13
Figura 2: Pizza 01 crua – gráfico do espaço de cores RGB da mesma pizza...	15
Figura 3: Pizza 01 após 4 minutos no forno – gráfico do espaço de cores RGB da mesma pizza	15
Figura 4: Pizza 01 após 8 minutos no forno – gráfico do espaço de cores RGB da mesma pizza	15
Figura 5: Pizza 01 após 12 minutos no forno – gráfico do espaço de cores RGB da mesma pizza	16
Figura 6: Pizza 01 após 16 minutos no forno – gráfico do espaço de cores RGB da mesma pizza	16
Figura 7: Pizza 01 após 20 minutos no forno – gráfico do espaço de cores RGB da mesma pizza	16
Figura 8: Pizza 01 após 24 minutos no forno – gráfico do espaço de cores RGB da mesma pizza	17
Figura 9– Foto original e as duas máscaras criadas a partir da imagem.....	18
Figura 10: mostra as informações fornecidas pela ferramenta Minitab para os dados do queijo da pizza 01, após 8 minutos de cocção para a variável R.	20
Figura 11: mostra as informações fornecidas pela ferramenta Minitab para os dados do queijo da pizza 01, após 8 minutos de cocção para a variável G.	21
Figura 12: Mostra as informações fornecidas pela ferramenta Minitab para os dados do queijo da pizza 01, após 8 minutos de cocção para a variável B.	21
Figura 13: Gráfico mostra a evolução dos valores de R, G e B da massa da pizza durante o processo de cocção.	22
Figura 14: resultado do grupo de cores RGB após a extração dos outliers.	25
Figura 15: gráfico da curva normal	28
Figura 16: fotos das pizzas utilizadas para a verificação do sistema, em ordem de cima para baixo, da esquerda para a direita, P1, P2, P3 e P4.....	29

Lista de Tabelas

Tabela 1: Valores Médios de R, G e B da população de referência	27
Tabela 2: valores de R, G e B das pizzas utilizadas para teste.....	28

1 Resumo

O método desenvolvido no presente trabalho pretende ser utilizado como um sistema de controle de qualidade não-supervisionado para a cocção de pizzas. Será utilizada uma metodologia Bayesiana para a resolução do problema. Nessa abordagem uma modelagem estatística preliminar será realizada como uma análise de dados explicativa dos padrões assados e crus. Em seguida, um experimento do monitoramento foi realizado com o objetivo de observar a evolução dos padrões, indo de cru à assada. Após a modelagem estatística, novas amostras serão checadas para verificar se as mesmas alcançaram a condição de cocção. O experimento concluiu que a metodologia pode ser utilizada como uma ferramenta de monitoramento para controle de qualidade.

2 Introdução

Apesar do grande avanço da tecnologia de automação no processo de produção industrial, pouco se faz nesta área quanto ao controle de qualidade na indústria de alimentos. Em grande parte dos casos esse controle é feito de maneira supervisionada por funcionários treinados que se encarregam de observar um produto, em sua fase final de produção, e verificar se ele satisfaz as condições mínimas de qualidade exigidas. Vários fatores podem contribuir para que ocorra a perda da eficiência no critério de avaliação supervisionada do produto. Por exemplo, o cansaço mediante a grande quantidade de amostras analisadas, a monotonia da atividade de avaliação repetitiva, o *stress* após longos períodos de trabalho e a análise subjetiva em função da preferência pessoal podem ser fatores de variabilidade e de qualidade na hora da seleção de um produto. A análise subjetiva do avaliador entre avaliadores diferentes também contribui para uma grande variabilidade do controle de qualidade das pizzas produzidas. A inspeção visual ou supervisionada além de ser extremamente cara, é de pouca confiança devido ao nível de atenção do funcionário supervisor diminuir com o tempo devido ao cansaço visual. Outros procedimentos além da inspeção visual, como tocar na borda de um produto, além de danificar alimentos delicados podem introduzir contaminação na cadeia produtiva.

Poucos casos de avaliação automatizada existem disponíveis na literatura pesquisada por se tratar de um processo industrial. Especula-se que tais procedimentos possam ser considerados como segredo industrial por parte das indústrias de alimentos. Os trabalhos na literatura em alguns casos são similares entre si e não possuem os mesmos objetivos propostos no presente trabalho.

Yan e Papadakis 2004 utilizaram um método simples no controle de qualidade na massa de pizzas assadas em microondas. Neste método a pizza depois de cozida era fotografada e a cor analisada qualitativamente e quantitativamente no PHOTOSHOP em três espaços de cores diferentes CMYK¹, RGB² e L* a* b*³. Neste trabalho os autores limitaram-se a uma simples análise da qualidade de cores e manchas do aspecto das pizzas depois de cozidas. Nenhum aspecto relativo ao controle de qualidade automático foi abordado.

Sun e Brossnam 2003 parte 1 utilizaram um sistema de visão computacional com lógica nebulosa para a automatização da checagem de qualidade da distribuição do molho de tomate na montagem de pizzas. No método em questão foi feita a análise do formato da massa e da distribuição do molho de tomate sobre a superfície. Com esta abordagem os autores conseguiram através de classificador proposto um acerto de 87% comparado à inspeção de qualidade realizada supervisionadamente.

Sun e Brossnam 2003 parte 2 utilizaram o mesmo sistema citado anteriormente, porém, nessa segunda parte do trabalho, analisou-se a

¹ CMYK = Espaço de cores Cyan, magenta, yellow e Black

² REG = Espaço de cores Red Green e Blue

³ L* a* b* = Espaço de cores lightness, a* Green to red chromatic component, b* blue to yellow chromatic component

disposição da cobertura da pizza, formada pelo queijo, o presunto, as azeitonas e o orégano. O sistema utilizou a mesma abordagem, analisando as imagens das pizzas, extraindo as informações necessárias e classificando as mesmas utilizando lógica nebulosa. O sistema conseguiu um acerto de 76% quando foram utilizadas cinco categorias de pizzas diferentes e um acerto de 100% quando foram utilizadas duas categorias.

Abdullah 2002 investigou o uso da tecnologia de visão computacional para classificar muffins baseado no aprendizado de cores utilizando um sistema computacional. Um dos objetivos principais dessa pesquisa era utilizar imagens para inspeção automática de muffins e outros produtos de panificação. No experimento era executada a inspeção visual do topo do muffin através de um colorímetro digital da marca Minolta. O colorímetro se aproximava espectralmente das funções do olho humano no espaço de cores $L^*a^*b^*$. Baseado nessas medidas conseguiu-se categorizar os muffins em três categorias distintas refletindo três graus de cocção: pouco assado, assado moderadamente e muito assado.

Du e Sun 2005 compararam três métodos de classificação de cobertura de pizza utilizando diferentes espaços de cores. Foram utilizados o RGB normalizado, HSV, I1I2I3, CIE $L^*a^*b^*$ e YCbCr. Como o método de classificação foi utilizado o algoritmo C4.5, redes neurais artificiais e o support vector machine (SVM). Os autores desconsideraram em cada um dos espaços de cores uma das variáveis envolvidas, ocasionando assim uma redução na dimensionalidade dos espaços de cores. A taxa de acerto variou entre 86,7% e 96,7%, tendo o melhor índice de acerto o classificador SVM com o espaço de cores HSV.

Outra forma de se fazer essa análise é através de instrumentos de medidas de cor. Porém, estes equipamentos exigem que se faça uma análise ponto a ponto da superfície do alimento. Este processo que é muito extenso e exige muito tempo de análise. Muitos desses instrumentos exigem que as amostras sejam homogeneizadas, através de um liquidificador ou moedor para se conseguir uma mistura homogênea. A homogeneização destrói a amostra, inutilizando-a para consumo. Outros sistemas de análise microscópica de imagens e cor também podem ser utilizados, porém são ineficientes na análise de objetos maiores, como pizzas e pães.

A produção em série, a melhoria nos padrões de higiene e qualidade dos alimentos através da automatização das linhas de produção gerou uma demanda por sistemas de avaliação objetiva e automática. Criando a necessidade de um controle de qualidade, baseado em parâmetros visuais como cor, tamanho e formato. Sistemas de visão computacional proporcionam soluções que podem atender a esta demanda. A partir desta necessidade da indústria, concluiu-se que um sistema de controle de qualidade automatizado seria de grande utilidade para este segmento industrial.

A partir de um convênio com a firma Bislau financiado por Petrobrás/Bister esta sendo desenvolvido um forno para pizzas utilizando a tecnologia de tubos de calor no LABTUCAL (Laboratório de Tubos de Calor – EMC/UFSC). Também no âmbito deste convênio, esta sendo desenvolvida uma tese de mestrado na área de Engenharia de Alimentos para analisar a qualidade das pizzas que estavam sendo produzidas na nova tecnologia de cocção. Este ambiente de desenvolvimento acabou gerando a necessidade de se desenvolver uma metodologia sistemática para analisar a qualidade das

pizzas produzidas no processo de cocção. Assim de uma forma conjunta optou-se por desenvolver uma metodologia de controle de qualidade durante a pesquisa que envolve aspectos sensoriais.

O presente trabalho se propõe a verificar a viabilidade da utilização da metodologia Bayesiana para a automatização do processo de controle de qualidade para a cocção de pizzas. O modelo prevê a definição de um modelo cognitivo a partir de uma análise sensorial das pizzas. Em seguida será feita uma caracterização estatística deste modelo no espaço de cores. Depois será implementado um teste estatístico de hipóteses para avaliar se a pizza esta pronta ou não de acordo com os padrões cognitivos estabelecidos. Nas próximas subseções serão descritos o aparato experimental utilizado, a análise supervisionada, a estatística da evolução do cozimento da pizza e os resultados encontrados.

3 Procedimento Utilizado

O procedimento geral utilizado na metodologia proposta para o controle de qualidade de alimentos será apresentado nas seguintes fases.

Obtenção das imagens de forma sistemática durante a cocção

Análise sensorial (para definição do ponto de cocção das pizzas)

Caracterização estatística dos padrões no espaço de cores, variabilidade destes padrões ao longo do processo de cocção.

- Eliminação de padrões expúrios (mascara de pixels)
- Extração das características principais no espaço de cores
- Caracterização final

Proposta do teste z para a Classificação

As estratégias e ferramentas adequadas para desenvolver a metodologia apresentada no presente trabalho serão explicadas no decorrer das próximas subseções.

3.1 Aquisição das imagens

Nesta fase do procedimento procurou-se controlar rigorosamente as condições de realização das fotos, para que fossem obtidas de forma sistemática com a menor alteração possível entre si. Evitando assim variações nas condições experimentais que poderiam introduzir erros e comprometer os resultados das fases subseqüentes.

As imagens das pizzas analisados no presente experimento foram obtidas a partir de uma câmera digital, modelo Cannon EOS Digital Rebel. As condições de iluminação foram controladas e padronizadas. Como fontes de iluminação, foram utilizadas duas lâmpadas de 6500 K, também utilizadas para minimizar as variações da luz ao longo do experimento. Essas lâmpadas, medindo 45 cm cada uma, foram fixadas a uma altura de 40 cm da base e

distantes 50 cm entre si. O ângulo formado entre o eixo da lente da câmera e cada uma das fontes de iluminação foi de aproximadamente 45°. A distância entre a lente da câmera e a pizza foi de 30 cm. As imagens da superfície da pizza foram obtidas sob as seguintes condições de operação da câmera: representação de cores RGB, comprimento focal de 28 mm, abertura do diafragma de F/5,6, tempo de exposição 1/200 segundos, método padrão de medição, utilização de flash. As imagens foram obtidas no formato JPG padrão da câmera. A resolução inicialmente escolhida de 4 megapixels precisou ser reduzida para diminuir o tempo de processamento, a complexidade da análise estatística e o tratamento das imagens.

Para a caracterização estatística foram utilizadas durante o experimento três pizzas sabor mussarela da linha Appreciatta da marca Perdigão. As pizzas foram assadas em um forno elétrico da marca Fischer ajustado para 270°C. Todos os dados de temperatura foram adquiridos em intervalos regulares de tempo, através do sistema de aquisição de dados HP 34970. O termopar utilizado na medida da temperatura foi o de tipo K. Para a obtenção das fotos durante o processo de cocção foi adotado o seguinte procedimento.

A primeira foto obtida durante todo o processo de cocção era com a pizza crua e descongelada, em seguida a pizza era levada ao forno para ser assada. A pizza era fotografada em tempos de cocção que eram múltiplos de 4 minutos. Quando era chegada a hora de tirar a foto, a porta do forno era aberta e a pizza retirada para que a sua foto pudesse ser obtida.

Enquanto a foto era tirada, a porta do forno permanecia fechada e a temperatura subia de forma constante. Aguardava-se que a temperatura dentro da câmara do forno chegasse novamente a 270°C para a porta ser aberta e a pizza inserida. Caso a foto não ficasse dentro dos padrões esperados (fora de foco, escura, posição ruim), uma nova foto era realizada. Durante todo esse processo, somente foi necessário repetir o processo de obtenção de imagem de uma mesma pizza uma única vez, que foi com a primeira pizza e serviu para calibrar o foco da máquina fotográfica. Esse processo ocorreu de forma constante até que a pizza ficasse totalmente queimada, quando não era mais possível verificar de forma visual uma mudança na cor da superfície da mesma. Esse tempo foi de vinte e quatro minutos para todas as pizzas analisadas. Ao final do processo, tínhamos como resultado uma seqüência de sete fotos para as três pizzas, sendo que para todas as pizzas foi produzida uma seqüência de fotos cujo tempo inicial foi de zero minuto, com a foto da pizza crua e o tempo final foi de vinte e quatro minutos, com a pizza totalmente queimada, resultando assim em uma seqüência com tempos de 0, 4, 8, 12, 16, 20 e 24 minutos. O total de fotos durante esta fase do experimento foi de vinte e uma.

A temperatura de cozimento do forno foi estabelecida em 270 °C, mas com a constante abertura e fechamento da porta para obtenção das fotos a temperatura oscilava, mas em seguida subia novamente a 270 °C.

Ao se abrir a porta do forno a temperatura dentro da câmara de cocção caía e alcançava valores entre 220° e 230°C. Após o fechamento da porta do forno, a temperatura dentro da câmara subia de forma constante até aproximadamente 270°C. Na média a pizza assou a uma temperatura de aproximadamente 250°C.

A figura 01 mostra a variação de temperatura durante o processo de cocção das três pizzas utilizadas no experimento. Nota-se que durante a

cozção da primeira pizza houve um comportamento irregular entre dezesseis e vinte e oito minutos de análise. O mesmo se deve a um deslocamento do termopar localizado dentro da câmara do forno. Ao ser percebido, o mesmo foi colocado na posição correta e todo o processo voltou a ocorrer de forma constante.

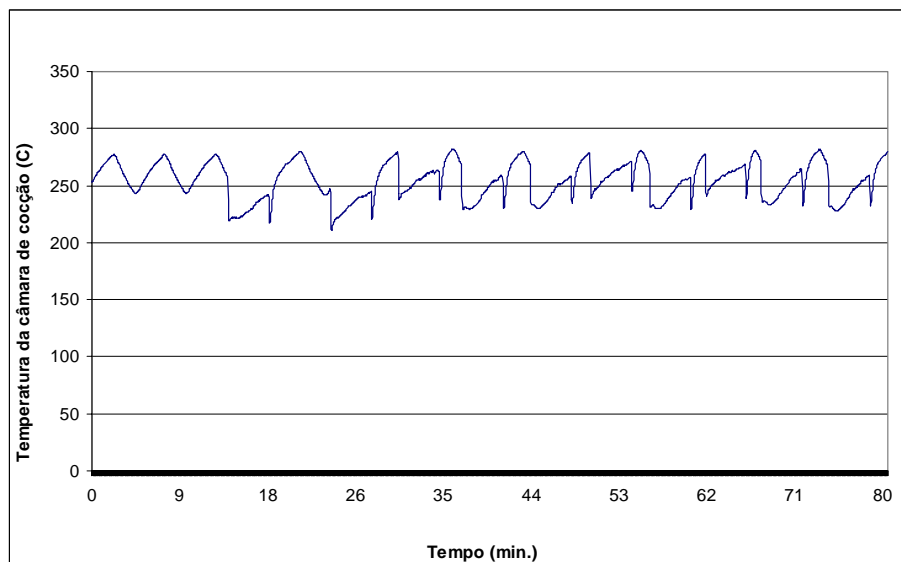


Figura 1: Curva de temperatura durante o processo de cozção

O manuseio da pizza foi realizado sempre com a utilização de luvas. A superfície onde a pizza era colocada para que a foto pudesse ser realizada era mantida sempre limpa. Esse cuidado foi necessário para que a pizza não fosse contaminada com qualquer tipo de substância e não tivesse assim o seu processo de cozção alterado, resultando numa alteração na sua cor original.

3.2 Criação das Máscaras

Após a obtenção das imagens elas foram carregadas em uma ferramenta de mascaramento de pixels desenvolvida no LAPIX⁴ denominada GTG⁵. O GTG permitiu que elementos representados por padrões na imagem que não representavam os padrões de interesse fossem eliminados manualmente através de uma máscara.

Para a criação das máscaras das imagens era necessário determinar quais características ou fatores na imagem de uma pizza seriam levados em consideração, para determinar se uma pizza está em condições próprias para consumo, ou se a mesma assou demais ou se não foi assada o suficiente. Nas pizzas analisadas, a cobertura crua era formada por uma camada de queijo picado, molho de tomate, de três a cinco azeitonas pretas, e a massa da pizza propriamente dita.

⁴ Laboratório de Processamento de Imagens – INE UFSC

⁵ GTG Ground Truth Generator Tool

Em testes realizados anteriormente foi verificado que a cor da azeitona preta praticamente não se altera durante a cocção da pizza, logo, optou-se por descartar as azeitonas como fator de determinação de cocção.

O molho de tomate, apesar de sofrer alterações em sua cor durante o processo de cocção, representava uma parte insignificante na imagem da pizza. Na maioria dos casos, com o derretimento do queijo e o encolhimento da massa, a área do molho diminuía à medida que o tempo de cocção da pizza aumentava. Por causa disto, o molho de tomate foi descartado como fator para determinação do estado da pizza.

O queijo sofreu uma modificação constante tanto em sua cor como em sua aparência. O queijo picado da cobertura da pizza tendia a se tornar uma figura homogênea e a cor do mesmo escurecia conforme o processo de cocção se realizava, logo, o queijo foi um dos fatores escolhidos para determinar se uma pizza estava boa ou não.

A massa apresentou comportamento semelhante ao do queijo, pois, a mesma escurecia de forma constante enquanto era assada, sendo assim, a massa também foi escolhida como um fator de determinação. Para a análise do posicionamento dos padrões no espaço de cor, o aplicativo Colorinspector3d⁶ foi utilizado.

A seqüência de imagens da figura 01 até a figura 08 evidencia bem como o nível de cores da pizza se alterou ao longo do processo de cocção. Existem três áreas bem definidas no gráfico de cores. A primeira área mais negra, fica próxima do valor zero (ou origem) do gráfico e representa em sua grande maioria as azeitonas e uma pequena parcela escura do fundo da imagem. A segunda área mais clara do gráfico representa em sua totalidade o fundo branco da imagem e se localiza no canto superior direito do gráfico. A área maior representa a massa, o queijo e um pouco do molho da cobertura da pizza. Nota-se com o auxílio dos gráficos a evolução da tendência que a massa e o queijo têm para o escurecimento ao longo do processo de cocção. Pode-se verificar uma mudança gradual no comportamento da área maior do gráfico. Essa mesma figura sofre uma redução em direção ao eixo da cor vermelha (R) da figura, mostrando assim uma diminuição no seu componente de intensidade. É verificado também um aumento significativo na área mais escura da figura. Isso ocorre devido ao escurecimento que a borda da massa da pizza sofre ao longo do processo de cocção. Conforme foi verificado nesse teste e em outros, a borda da massa tem a tendência de escurecer de forma mais acentuada do que o restante da massa da pizza.

⁶ <http://rsbweb.nih.gov/ij/index.html>

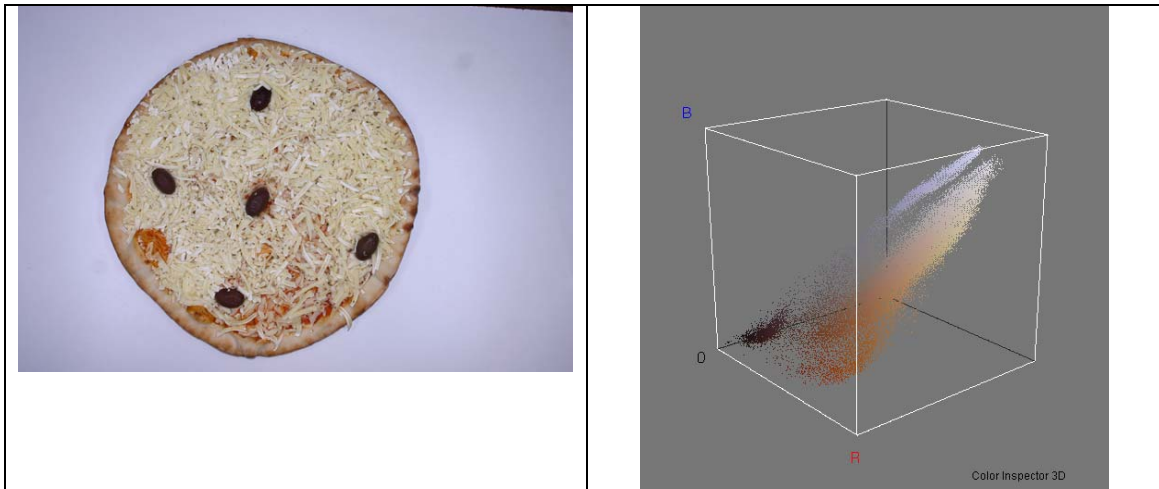


Figura 2: Pizza 01 crua – gráfico do espaço de cores RGB da mesma pizza

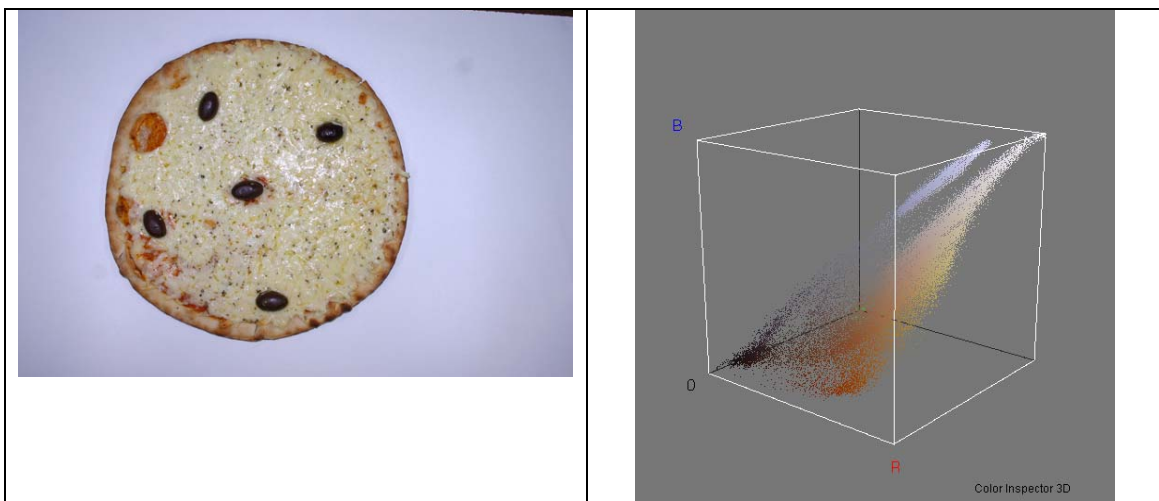


Figura 3: Pizza 01 após 4 minutos no forno – gráfico do espaço de cores RGB da mesma pizza

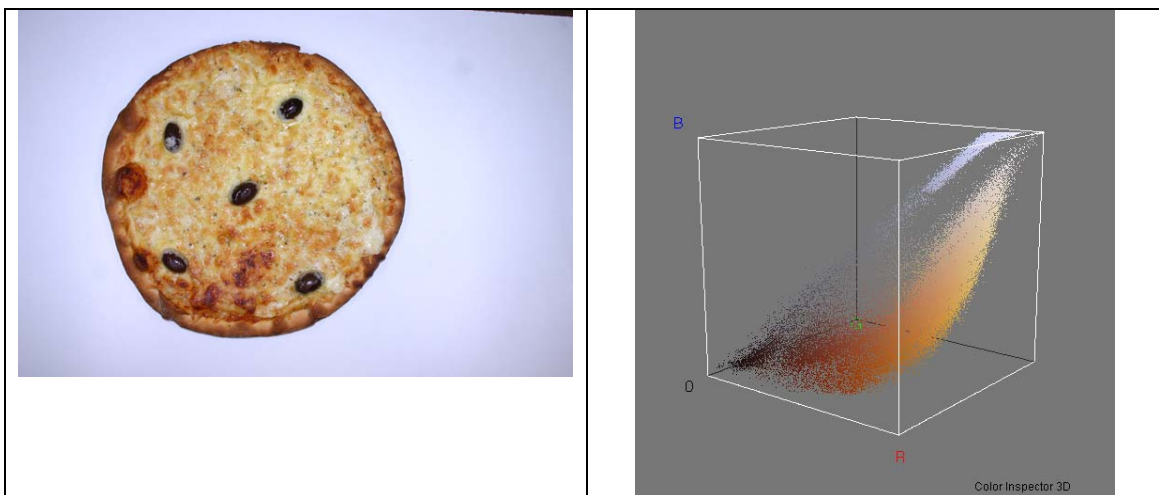


Figura 4: Pizza 01 após 8 minutos no forno – gráfico do espaço de cores RGB da mesma pizza

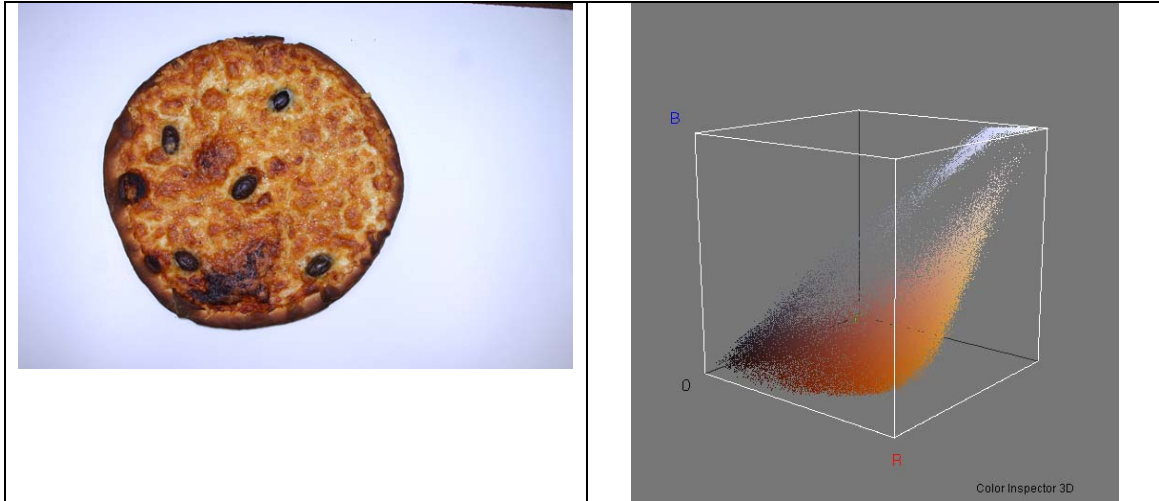


Figura 5: Pizza 01 após 12 minutos no forno – gráfico do espaço de cores RGB da mesma pizza

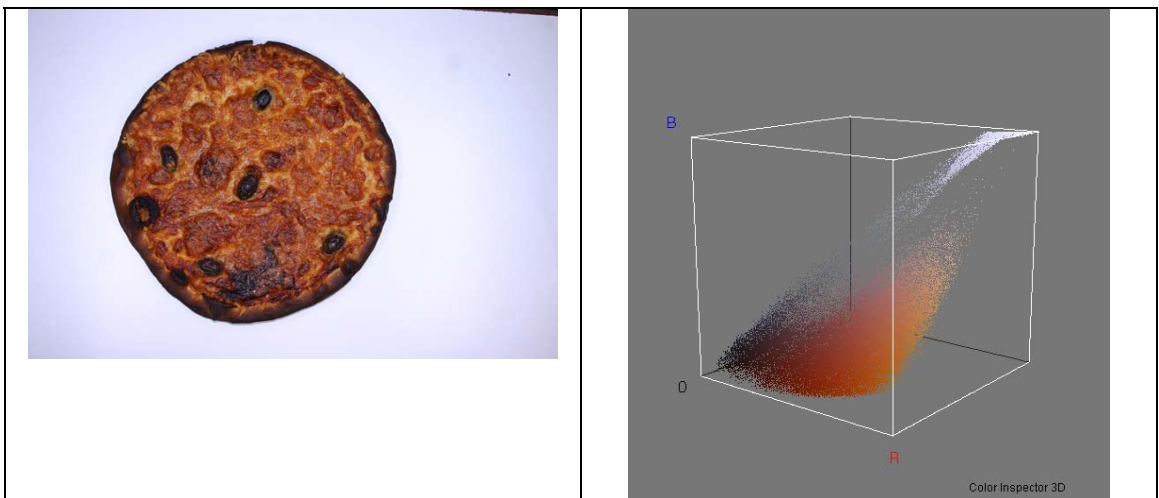


Figura 6: Pizza 01 após 16 minutos no forno – gráfico do espaço de cores RGB da mesma pizza

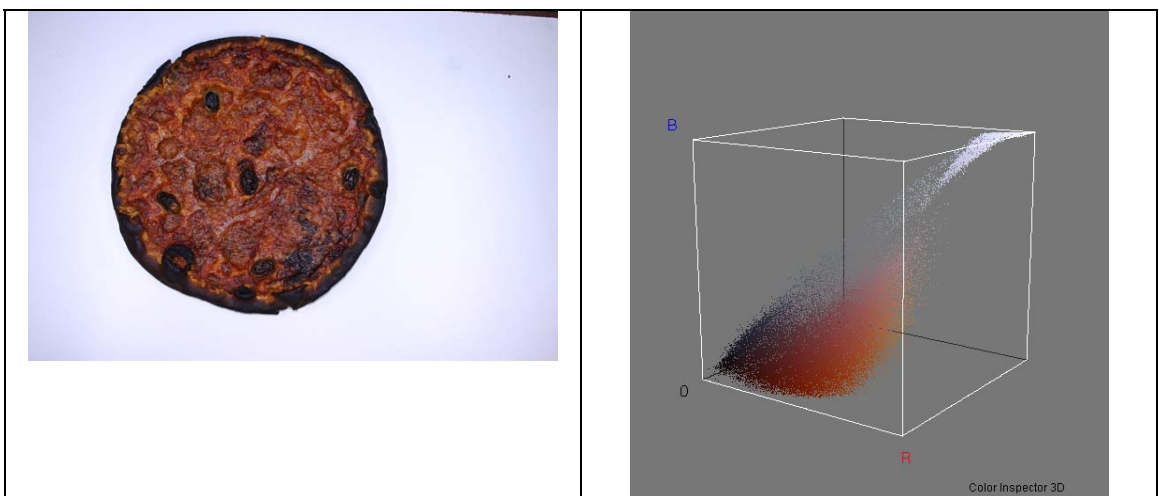


Figura 7: Pizza 01 após 20 minutos no forno – gráfico do espaço de cores RGB da mesma pizza

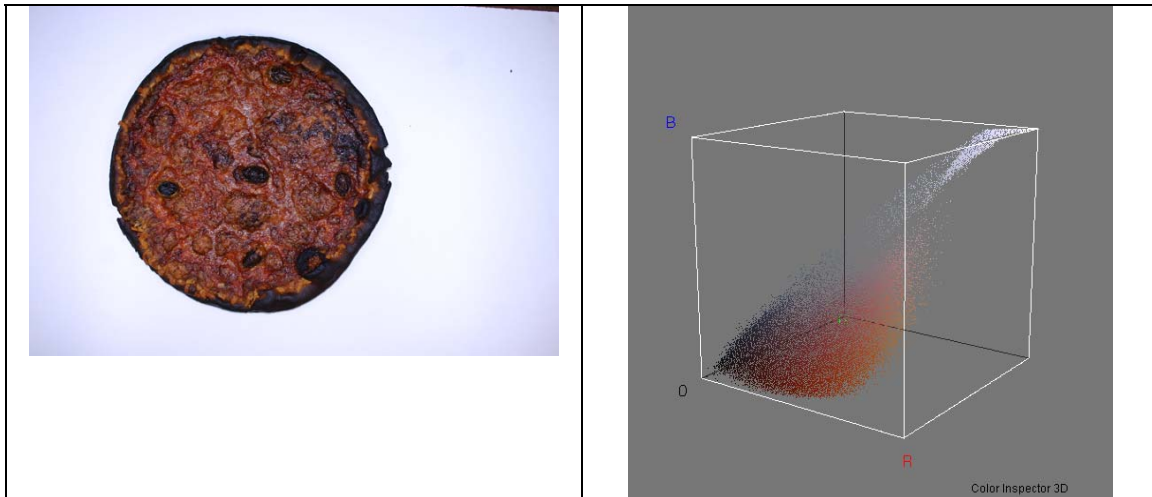


Figura 8: Pizza 01 após 24 minutos no forno – gráfico do espaço de cores RGB da mesma pizza

Uma vez definidas as características da pizza que seriam analisadas, restava somente isolar os padrões (queijo e massa) dos arquivos de imagens para que as informações pudessem ser analisadas estatisticamente. A separação foi feita de forma manual baseado na análise sensorial. Apesar de ser um processo relativamente lento, a segmentação manual é um processo de resultado garantido, pois representa fielmente os padrões desejados durante a análise sensorial conhecido como Ground Truth.

Para a realização do processo de segmentação da imagem foi utilizado o software GTG. Esse software foi desenvolvido pela equipe do projeto Cyclops utilizando a linguagem Delphi com o objetivo de se segmentar imagens digitais de forma manual. A utilização desse software é bastante simples. É necessário fazer uma seleção fechada de uma área da imagem selecionada. Seleção fechada significa que o ponto inicial da imagem é igual ao ponto final. Essa área de seleção é feita à mão livre, sendo que pode ter qualquer formato, tornando assim esse processo de seleção bastante livre e eficiente. A ferramenta permite também que o usuário selecione mais de uma área da imagem, de forma separada. Ao término da seleção das áreas, determina-se qual ou quais delas serão transformadas em preto, ou seja, terão os seus valores de R, G e B atualizados para zero.

A separação dos padrões das imagens foi feita de modo diferente tanto para a massa quanto para o queijo. O processo de separação do queijo foi mais simples. Somente foi necessário selecionar a área onde o queijo se encontrava com a massa, ignorar a mesma, as azeitonas pretas e descartar qualquer vestígio de molho que fosse encontrado. Neste processo também, foi descartado todo o fundo branco da imagem. O processo de separação da massa foi feito através da borda da massa, para que pudesse ser descartado, o fundo, toda a área do queijo, incluindo as azeitonas, qualquer sinal de molho e de queijo que por ventura tivesse sido encontrado sobre a massa. Para cada uma das imagens da pizza foi criada tanto uma máscara para a massa quanto para o queijo, resultando ao final do processo quarenta e duas imagens marcadas. Metade dessas imagens correspondia à seleção da massa das imagens e a outra metade correspondia ao queijo. A figura 09 mostra um exemplo de uma imagem que foi segmentada em massa e queijo.



Figura 9– Foto original e as duas máscaras criadas a partir da imagem.

3.3 Extração dos Valores dos Pixels

Tendo em mãos as máscaras das fotos da pizza era necessário agora extrair as informações dos pixels da máscara da pizza. Os dados necessários para análise das imagens foram os valores de R, G, B, H, S, L e a posição do pixel na foto. O valor do pixel é representado pelas letras x para a posição horizontal e y para a posição vertical. Os valores de R, G e B puderam ser extraídos diretamente da imagem. Mas os valores de H, S e L deveriam ser calculados para cada pixel da imagem, a partir dos respectivos valores de R, G e B para o pixel em questão.

Esses valores foram extraídos utilizando-se um algoritmo que foi especialmente desenvolvido pelo graduando para a realização da tarefa. Para a implementação do mesmo foi utilizada a linguagem C++, por ser utilizada já de forma consistente e com sucesso em aplicações envolvendo visão computacional e por ter disponível uma grande variedade de bibliotecas para essa mesma área de atuação. Várias bibliotecas foram analisadas, porém, a escolhida foi a OpenCV, por ser de fácil utilização e possuir uma documentação bastante completa disponível na internet. O código fonte encontra-se no anexo A.

“A biblioteca OpenCV (*Open Source Computer Vision Library*) originalmente desenvolvida pela Intel, em 2000, é uma biblioteca multi-plataforma, totalmente livre ao uso acadêmico e comercial, para o desenvolvimento de aplicativos na área de Visão Computacional, bastando seguir o modelo de licença da BSD Intel. O OpenCV possui módulos de Processamento de Imagens e Vídeo I/O, Estrutura de dados, Álgebra Linear, GUI (Interface Gráfica do Usuário) Básica com sistema de janelas independentes, Controle de mouse e teclado, além de mais de 350 algoritmos de Visão Computacional como: Filtros de imagem, calibração de câmera, reconhecimento de objetos, análise estrutural e outros. O seu processamento é em tempo real de imagens.

Esta biblioteca foi desenvolvida nas linguagens de programação C/C++. Também, dá suporte a programadores que utilizem Python e Visual Basic e desejam incorporar a biblioteca a seus aplicativos. A última versão disponível é a 1.0, lançada no final de 2006” (Wikipédia 2008).

O algoritmo funcionava da seguinte maneira: ele percorria a imagem do ponto de origem (0,0), que se encontrava no canto superior esquerdo da imagem e caso algum valor de R, G ou B fosse igual à zero, o mesmo era descartado, pois, se tratava de um valor indexado artificialmente durante a criação das máscaras, não fazendo parte da pizza da imagem em questão.

Quando o algoritmo encontrava um valor diferente de zero o mesmo armazenava os valores de R, G e B e utilizava os mesmos para calcular os valores de H, S e L e gravava todos os valores citados, mais a posição (x, y) do pixel em um arquivo de extensão.txt.

OpenCV não representa a imagem como uma matriz, a imagem é representada como um array. Ex: caso a figura tenha 100 pixels e esteja utilizando o padrão de cores RGB, a mesma figura em Opencv será representada por um array de 300 pixels e o padrão de cores será o BGR. As imagens em OpenCV não são armazenadas por pixels. As mesmas são armazenadas em um array de nível de cores que leva a um overhead de processamento menor já que não se está dereferenciando ponteiros de forma constante. Esses arrays de cores são guardados na ordem BGR. Não foi encontrada a razão pela qual os valores dos níveis de cores estão na ordem BGR e não no tradicional RGB. OpenCV trabalha de forma diferente com os pixels, os valores de R, G e B do pixel (0,0) se encontram nas posições (0 para B), (1 para G) e (2 para R). Segue abaixo a representação dos padrões de cores em OpenCV e o tradicional RGB.

imageData[0]	imageData[1]	imageData[2]	imageData[3]	imageData[4]	imageData[5]
B	G	R	B	G	R

Os campos de um objeto da classe IplImage são representados da seguinte maneira...

imageData[0]			imageData[1]		
->R	->G	->B	->R	->G	->B

Em oposição ao formato comum.

Assim, como resultado desse estágio, produziu-se quarenta e dois arquivos de extensão.txt, um para cada máscara produzida. Esses arquivos foram salvos de forma que os dados de um pixel referentes à sua posição na imagem, seus valores de R, G, B, H, S e L estavam dispostos em uma mesma linha e esses mesmos dados eram separados por um espaço simples. Esse formato permitia que o arquivo fosse importado de forma direta pela ferramenta de análise estatística, que será mais bem explicada no próximo estágio.

3.4 Análise dos valores e determinação do intervalo

Depois que os valores dos píxeis correspondentes aos padrões de queijo e massa foram separados, foi feita a caracterização estatística dos padrões ao longo do processo de cocção. Para isto foram utilizados valores dos píxeis convertidos para arquivos txt no Minitab. “O Minitab é um programa de computador proprietário voltado para análise estatística. com capacidade para analisar uma quantidade de dados muito maior que por exemplo o Microsoft Excel. O Minitab importou os arquivos com essas características de forma direta, não precisando fazer nenhuma modificação. Todos os quarenta e dois arquivos de texto gerados na fase anterior foram importados, gerando seis projetos na ferramenta onde cada projeto continha sete planilhas com oito colunas de informações sendo que três projetos continham os dados da massa e os outros três continham os dados do queijo.

Após a ferramenta ter sido alimentada com os dados, começou-se a análise dos valores. Para cada planilha determinou-se os valores médios das variáveis envolvidas, assim como os valores dos primeiro e terceiro quartis da mesma. Esses valores eram calculados de forma direta pela ferramenta e a mesma disponibilizava uma análise estatística completa da variável envolvida. Entre esses valores se encontrava a curva de distribuição. Essas curvas demonstraram que durante a fase de separação dos dados da imagem houve uma sobreposição de valores de RGB nas regiões por onde foi feita a seleção com o GTG. Nessas regiões os valores ficaram próximos de zero e não causaram uma pequena distorção na curva de distribuição, porém, na segunda parte dessa etapa os mesmos foram todos excluídos com o método que será explicado no transcórre dessa etapa. As figuras 10, 11 e 12 são um exemplo de um dos inúmeros gráficos produzidos durante essa fase do processo. Os gráficos em questão contem informações que serão utilizadas no processo de classificação. Entre os valores disponibilizados estão a média da variável, a mediana, os primeiro e terceiro quartis, sendo esses últimos valores utilizados para se definir a faixa de trabalho.

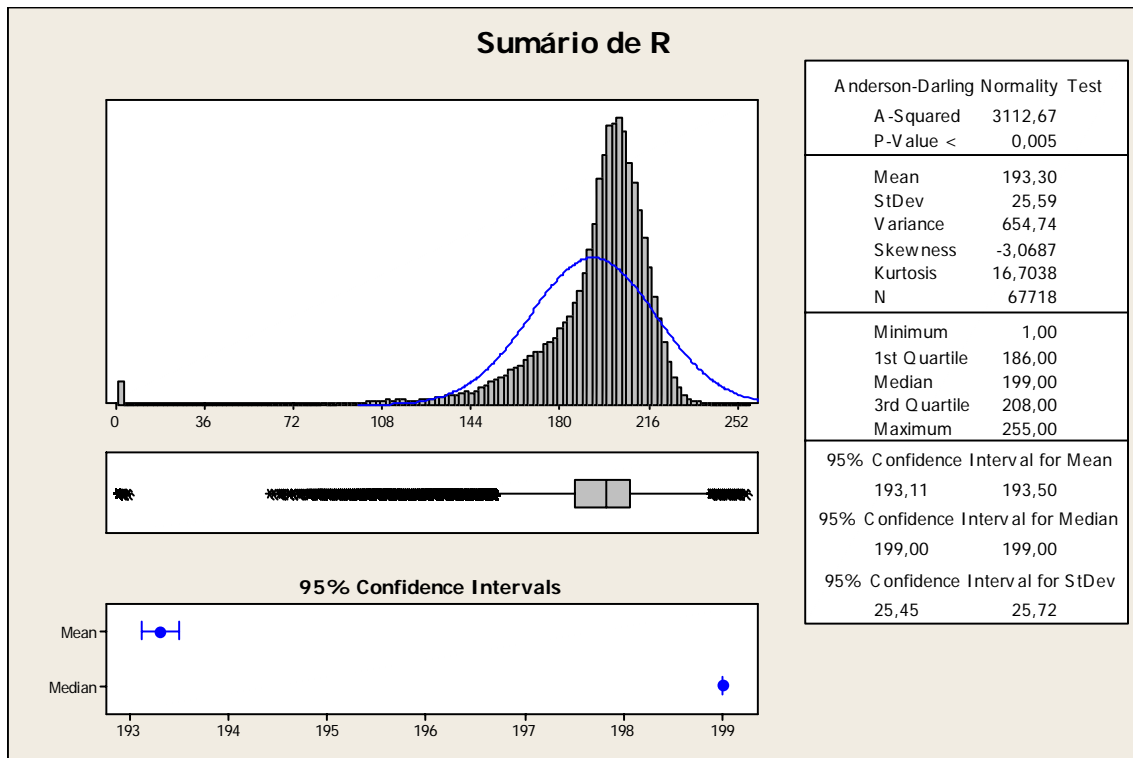


Figura 10: mostra as informações fornecidas pela ferramenta Minitab para os dados do queijo da pizza 01, após 8 minutos de cocção para a variável R.

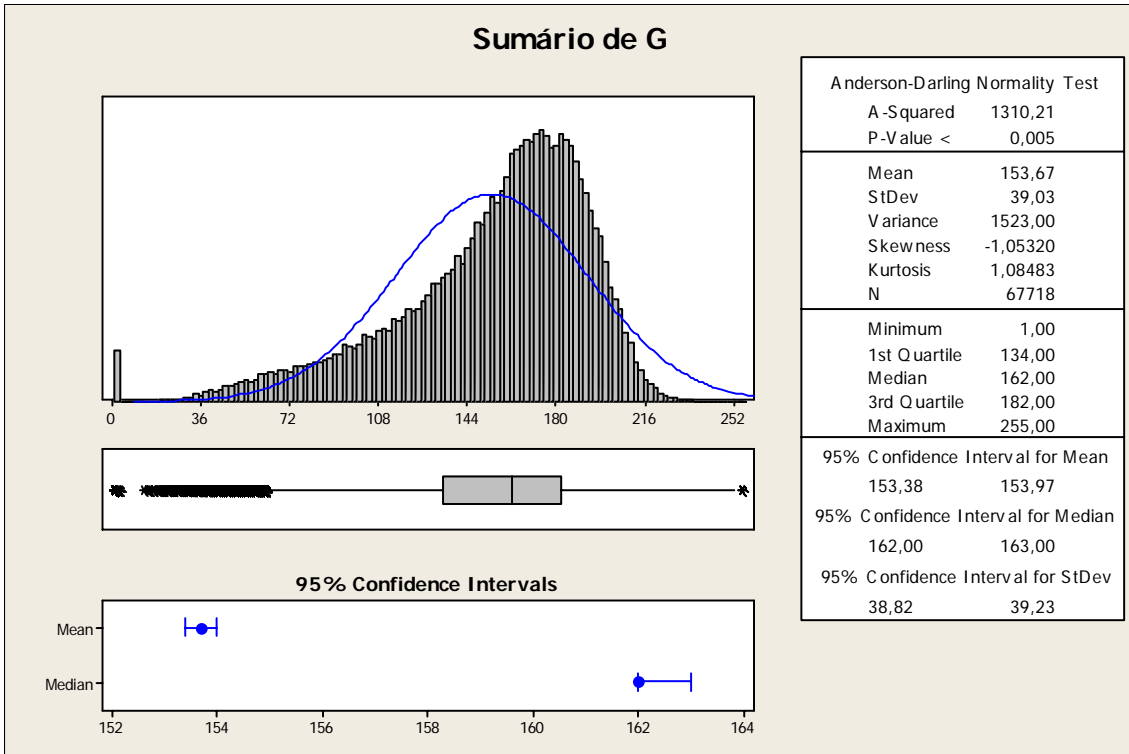


Figura 11: mostra as informações fornecidas pela ferramenta Minitab para os dados do queijo da pizza 01, após 8 minutos de cocção para a variável G.

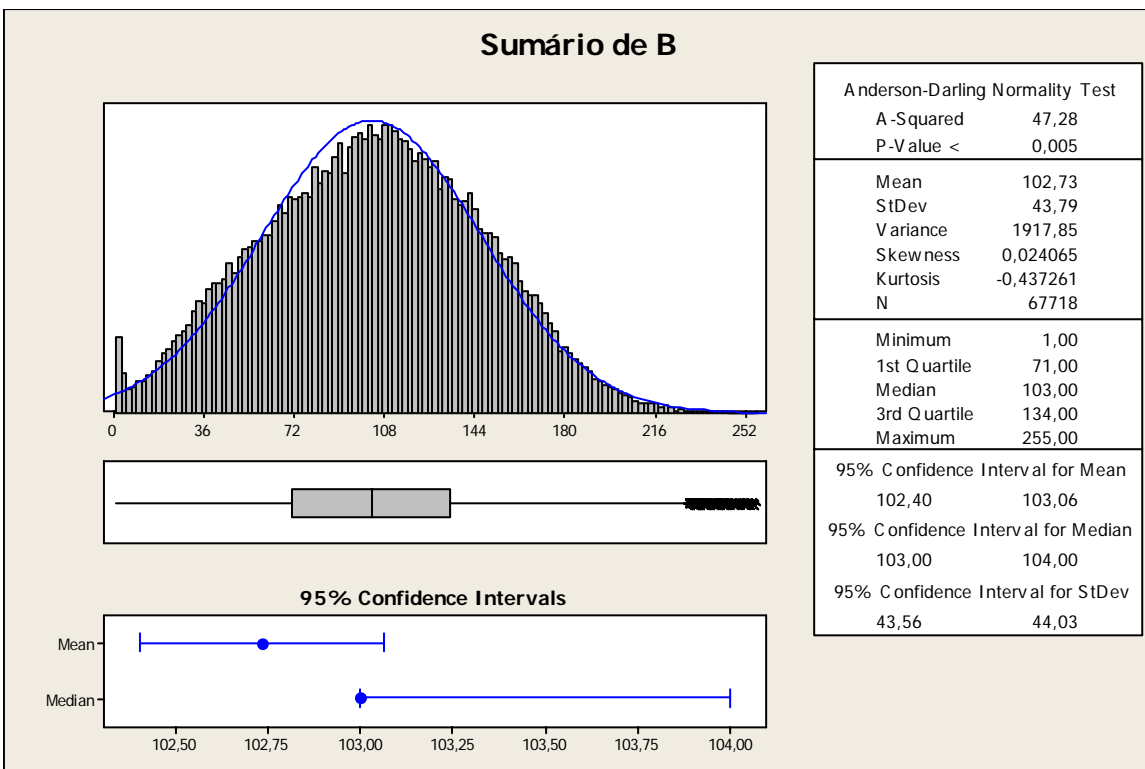


Figura 12: Mostra as informações fornecidas pela ferramenta Minitab para os dados do queijo da pizza 01, após 8 minutos de cocção para a variável B.

Tendo todos os valores das imagens da pizza na ferramenta, foi possível analisar o comportamento médio dos pixels das imagens das pizzas

ao longo de todo o processo de cocção. Os valores médios dos diferentes espaços de cores e das três pizzas foram salvos em uma nova planilha para que esses valores pudessem ser analisados. Para realizar a análise foram gerados vários gráficos da massa e do queijo separadamente nos dois espaços de cores utilizados, RGB e HSL. É possível ver através dos gráficos (ver figura 13) um comportamento semelhante das suas variáveis R, G e B entre as pizzas analisadas. Essa é uma forte evidência de que é possível se classificar uma pizza quanto ao seu estado de cocção utilizando visão computacional. Os valores médios dos grupos de cores resultantes dessa fase podem ser encontrados no anexo B.

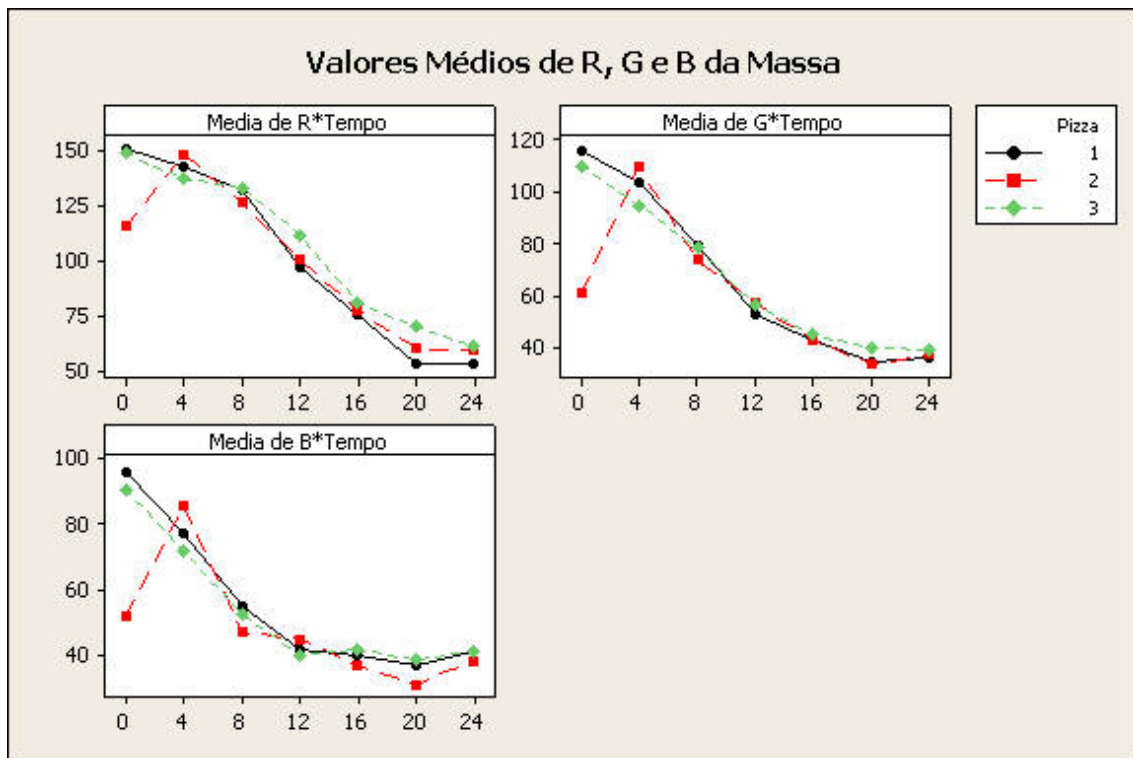


Figura 13: Gráfico mostra a evolução dos valores de R, G e B da massa da pizza durante o processo de cocção.

Após ter alimentado o sistema com os dados começou-se a analisar os dados de cada imagem. Foi possível observar nos gráficos gerados pelo minitab que houve um acúmulo de “dados suspeitos” ou outliers bem próximos do zero no gráfico de informação de todas as imagens. Um outlier ou dado suspeito é uma observação que se encontra fora do padrão médio de uma distribuição (Moore and McCabe 1999). Outliers são fáceis de ser detectados em histogramas. Uma definição conveniente de um outlier é um ponto que se encontra a uma distância 1,5 vezes maior que a distância inter quartis acima do terceiro quartil ou essa mesma distância abaixo do primeiro quartil. Usualmente, a presença de um outlier indica algum tipo de problema. Esses problemas podem ser de dois tipos, ou o outlier tem origem em um erro de registro; por exemplo, o valor real 136 foi registrado como 2136, nesse caso, o erro pode ser corrigido facilmente e a característica suspeita pode ser eliminada do dado amostrado. Outro problema pode ser que a amostra em questão faça parte de outro grupo amostral ou o mesmo demonstra a

ocorrência de um acontecimento extraordinário ou uma variabilidade não esperada da variável.

Foi verificado utilizando-se a ferramenta ColorInspector3D que no local das imagens onde foi feita a seleção das áreas houve uma sobreposição entre os valores da imagem da pizza com os valores 0, 0 e 0 para R, G e B que foram atualizados pela ferramenta GTG durante a fase de criação das máscaras. Assim, nessas regiões os valores de R, G e B ficaram bem próximos de zero, mas não igual, fazendo com que os mesmos não fossem descartados pelo algoritmo na fase de extração dos dados. Porém, como esses dados se encontram fora da curva de distribuição dos grupos de cores, os mesmos seriam descartados utilizando-se uma abordagem estatística, utilizando-se distância entre quartis. O colorInspector3D é uma ferramenta desenvolvida em java para a análise de imagens através de histogramas. Onze espaços de cores diferentes e cinco modos de visualização são suportados.

Em análise estatística a mediana, que divide a amostra em duas partes iguais, o primeiro quartil, que divide a primeira metade da mediana pela metade e o terceiro quartil, que divide a segunda metade da mediana pela metade. Essas três variáveis avaliam a forma da parte central e a variabilidade da distribuição de freqüências da amostra. A distância entre quartis ou IQD é o resultado da diferença entre o terceiro quartil Q3 e o primeiro quartil Q1:

$$\text{IQD} = \text{Q3} - \text{Q1}$$

As principais características da IQD são:

É uma medida simples e fácil de ser calculada.

Mede a distribuição da metade dos dados da amostra situados ao redor da mediana.

É uma medida resistente, pois não é afetado pelos dados extremos da amostra ou variável.

Não é suficiente para avaliar a variabilidade da amostra ou variável, pois envolve apenas os valores centrais, deixando de considerar os valores extremos que também são importantes, os restantes 50% dos dados.

Sendo assim, era preciso excluir os outliers do grupo de amostragem. Para cada planilha foi calculado para cada espaço de cor a distância entre quartis e foi calculado o intervalo que seria considerado válido. Conforme citado anteriormente, os outliers são pontos que se encontram a uma distância 1,5 vezes maior do que a distância entre quartis acima do terceiro quartil e a mesma distância abaixo do primeiro quartil. Segue abaixo o cálculo que foi feito para os valores de R do queijo da pizza 01 crua. O restante dos valores pode ser encontrado no anexo D.

Valores dos quartis:

3º quartil – Q3: 194

$$\text{IQD} = \text{Q3} - \text{Q1}$$

1º quartil – Q1: 175

$$\text{IQD} = 19$$

$$\text{Limite superior} = \text{Q3} + \text{IQD} * 1,5$$

$$\text{Limite Inferior} = \text{Q1} - \text{IQD} * 1,5$$

$$\text{Limite superior} = 194 + 28,5$$

$$\text{Limite Inferior} = 175 - 19 \cdot 1,5$$

$$\text{Limite superior} = 222,5$$

$$\text{Limite Inferior} = 146,5$$

Esses cálculos foram realizados utilizando-se um algoritmo em C++ para tornar o processo mais rápido, o mesmo pode ser encontrado no anexo C. Para cada um dos quarenta e dois conjuntos de dados esse mesmo cálculo foi realizado para cada espaço de cor, fazendo assim com que em cada espaço de cor analisado, todos os outliers fossem excluídos das distribuições, fazendo assim com que os gráficos das imagens não sofressem mais interferência desses valores e ficassem o mais próximo possível do comportamento real das variáveis das pizzas. Com os intervalos definidos, foi necessário testar de forma isolada cada pixel da imagem para determinar se o mesmo se encontrava dentro do intervalo válido ou se estava fora, tendo assim que ser descartado. Todo esse processo foi realizado novamente utilizando-se o Minitab. Utilizando a função Calc do minitab foi possível criar funções para fazer essa seleção. A forma era bastante simples, para cada pixel de um determinado valor, verificava-se se o seu valor era maior ou igual que o limite inferior e se o mesmo valor era menor ou igual ao limite superior. O resultado dessa comparação era salvo em uma outra coluna da ferramenta. Ex: para o valor R da pizza 01 crua, os valores dos pixels deveriam estar entre o limite inferior 146,5 e o limite superior 222,5. Caso o valor estivesse fora desse intervalo, o mesmo era gravado em uma outra coluna diferente. Esse cálculo foi realizado para todas as variáveis em todas as planilhas.

Tendo os resultados em mãos, fazia-se necessário selecionar quais pixels fariam parte do grupo da solução e quais ficariam de fora da mesma. Assim, fez-se um AND lógico entre os valores de R, G e B e outro entre os valores de H, S e L. Esse AND lógico se fez necessário, pois, foi considerado que para cada pixel pertencente ao grupo da solução, todos os valores de R, G e B deveriam estar dentro do intervalo estipulado na fase anterior, assim como todos os valores de H, S e L. Desse cálculo se resultou em quatro novos grupos de valores, sendo esses quatro grupos formados pelos valores de R, G e B aceitos no intervalo, um segundo grupo formado pelos valores aceitos de H, S e L, um terceiro formado pelos valores de R, G e B não aceitos e por último um grupo formado pelos valores não aceitos de H, S e L. A figura 12 mostra o resultado dessa etapa de forma gráfica. Todos os valores calculados nessa fase podem ser encontrados no anexo B.

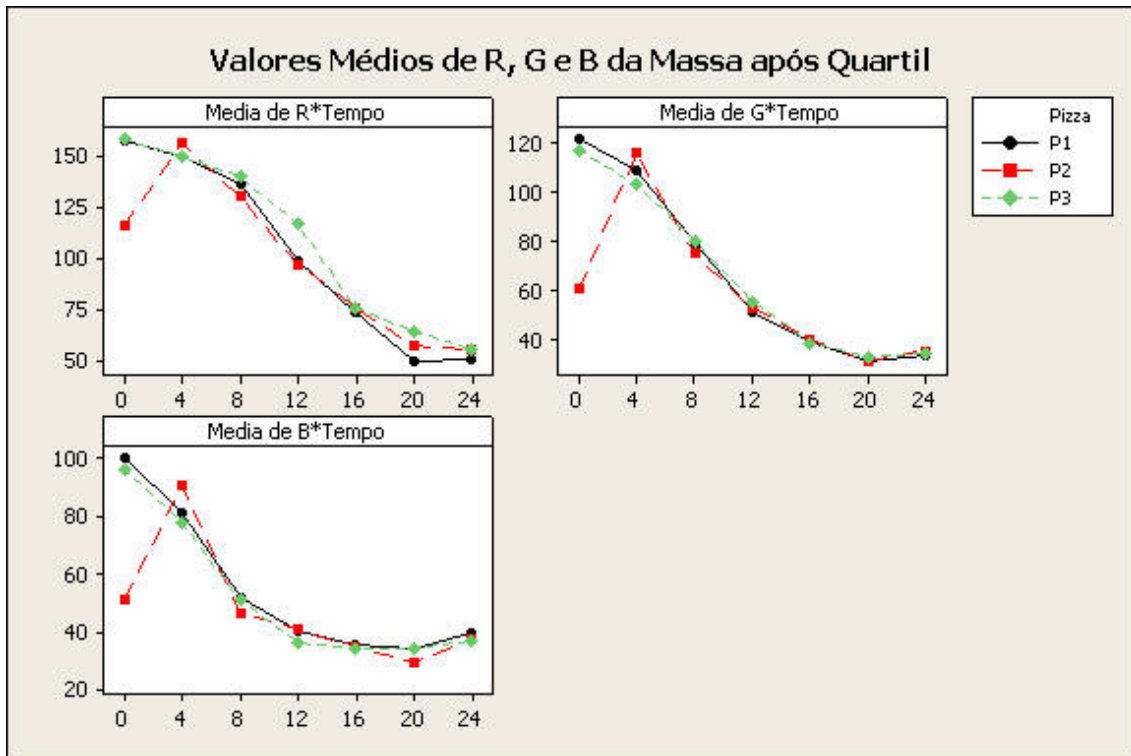


Figura 14: resultado do grupo de cores RGB após a extração dos outliers.

3.5 Análise sensorial

Essa é a fase final da metodologia proposta. Como resultado da fase anterior, temos as imagens das pizzas livres dos outliers, tornando assim as amostras mais homogêneas. Porém, para se conseguir classificar automaticamente uma pizza quanto ao seu nível de cocção, para conseguir dizer se a mesma está dentro dos padrões estabelecido pelos consumidores, é preciso primeiro conhecer esse padrão. Assim, foi realizada uma análise sensorial, para conseguir se determinar esse padrão de aceitação, para conseguir descobrir qual é a aparência da pizza que mais atrai as pessoas. Essa pesquisa foi conduzida por Bianca Sens dos Santos, mestranda do LABTUCAL desenvolvendo a sua Tese na área de Engenharia de Alimentos e responsável pelo desenvolvimento de um novo forno para pizza utilizando tubos de calor e termosifões.

Nessa pesquisa três atributos foram escolhidos para representar a qualidade da pizza, foram eles: a cor da pizza, textura da crosta e a elasticidade da massa. Porém, para esse trabalho, somente a cor da pizza foi considerada como fator de qualidade. Assim, a cor das pizzas foi avaliada pela comparação entre as imagens das pizzas assadas sob condições de cocção determinadas e classificadas segundo um escala de cor, que foi desenvolvida com essa finalidade.

Pizzas foram assadas em diferentes tempos: 0 (crua), 4, 8, 12, 16, 20, 24, 28, 32 e 38 minutos, no forno elétrico em seguida, fotografadas e codificadas. Os limites de aceitação e o nível ideal para a cor da pizza foram definidos através de análise sensorial, por teste de aceitação. Para este teste, foi utilizado o teste de escala Hedônica de 5 pontos, variando de “desgostei muito” a “gostei muito”. Sessenta e oito julgadores não-treinados, com idade

entre 20 e 45 anos, recrutada entre alunos e professores universitários, realizaram os testes, via correio eletrônico. Uma ficha para a realização dos testes foi desenvolvida e, juntamente com as imagens das pizzas, enviada aos julgadores. Os julgadores foram, então, solicitados a avaliar a cor das pizzas conforme a escala apresentada.

Através das respostas obtidas com essa pesquisa, foi possível determinar uma faixa de aceitação para as pizzas. Comparando-se os resultados da pesquisa com as imagens das pizzas que foram usadas ao longo desse trabalho, concluiu-se que as pizzas consideradas boas pela maioria das pessoas se encontram entre o intervalo de tempo de oito e dez minutos de cocção, quando assadas a uma temperatura média de 250°C.

Com o resultado da pesquisa, foi possível determinar a aparência de uma pizza considerada boa pela maioria das pessoas entrevistadas. Comparando o resultado da pesquisa com a população de pizzas analisadas, foi possível determinar que as pizzas dentro de uma faixa de tempo entre oito e nove minutos são consideradas próprias para consumo. Assim, iniciou-se a fase de determinação da média dos valores da população.

Devido à complexidade do próprio espaço de cor foi descartada a utilização do espaço de cores HSL. O mesmo, por ser representado de forma não linear tornava essa fase de comparação de valores muito mais complexa e devido a algumas inconsistências observada nos resultados (ver anexo B) foi descartado como métrica de comparação.

Para o cálculo das médias da população dos valores de R, G e B executou-se um processo bastante simples. Para cada valor das variáveis de cor da massa e da pizza foi realizado um cálculo de média convencional. Somou-se o valor de todas as amostras de uma variável e dividiu-se esse valor pelo número de amostras, determinando-se assim o valor médio das populações dos espaços de cores. Segue abaixo a fórmula utilizada para o cálculo da média:

$$X = \frac{\sum_{i=1}^n X_i}{n}$$

Também foi calculada a variância e o desvio padrão. A variância foi calculada utilizando-se a seguinte fórmula:

$$s^2 = \frac{\sum_{i=1}^n (X_i - X)^2}{n - 1}$$

O desvio padrão é a raiz quadrada da variância e foi calculado extraíndo-se esse valor. Os resultados desses cálculos são encontrados na tabela abaixo.

	Queijo			Massa		
	Média	Desvio Padrão	Variância	Média	Desvio Padrão	Variância
R	194,14	7,4847	56,02084	127,604	10,43	127,604
G	156,25	18,47	341,17	72,025	7,14	72,025

B	114,27	24,76	613,096	47,25	3,92	47,25
----------	--------	-------	---------	-------	------	-------

Tabela 1: Valores Médios de R, G e B da população de referência

Para a tentativa de classificação primeiramente tentou-se uma abordagem utilizando o Teste de Hipóteses. Esse método consiste em se determinar duas hipóteses independentes, ou seja, as duas não podem acontecer ao mesmo tempo, a intersecção das duas é vazia. As duas hipóteses seriam primeiramente a hipótese nula H_0 , que é a hipótese sobre a qual devem ser obtidas evidências para rejeitá-la e a hipótese alternativa H_1 , que é a hipótese sobre a qual devem ser obtidas evidências para aceitá-la. Nesse trabalho, a hipótese H_0 foi que a média da amostra era igual à média da população, $H_0: X = X_i$ e a hipótese H_1 era a de que a média da amostra era diferente da média da população, $H_1: X \neq X_i$. Como a quantidade da população era diferente da quantidade da amostra, foi necessário utilizar as seguintes fórmulas:

$$s^2 = \frac{(n1-1) * s1^2 + (n2-1) * s2^2}{n1 + n2 - 2} \text{ e } t = \frac{X1 - X2}{s * \sqrt{\frac{1}{n1} + \frac{1}{n2}}} \text{ onde:}$$

$s1^2$: variância da população

$s2^2$: variância da pizza analisada

$n1$: total de amostras da população

$n2$: total de amostras da pizza analisada

$X1$: média da população;

$X2$: média da pizza analisada

Porém, ao se utilizar a fórmula em questão, não foi possível de se realizar a classificação utilizando-se o teste de hipóteses. Esse problema foi causado pela quantidade das amostras da população de referência, que foi utilizada para o controle e a quantidade de amostras da pizza analisada, já que se estava considerando os pixels da imagem como a amostra. Segundo Robert S. Witte, “uma amostra excessivamente grande produz um teste de hipótese extremamente sensível, que detecta até o efeito menos importante”. Sendo assim, a utilização desse método foi descartada.

Assim, era necessária a utilização de outra metodologia, que não levasse em consideração o número de pixels das amostras. Foi utilizada então a distribuição normal para essa verificação. . Nessa abordagem, um valor z é calculado levando em consideração a média da população, a média da amostra e o desvio padrão da população. Segue abaixo a fórmula utilizada para o cálculo de z .

$$z = \frac{X_i - X}{s} \text{ onde:}$$

X : média da população

X_i : média da amostra

s : desvio padrão da população

Depois, procura-se em uma tabela padrão a probabilidade desse valor z se encontrar abaixo da curva normal. A figura xxx mostra a imagem de uma curva normal.

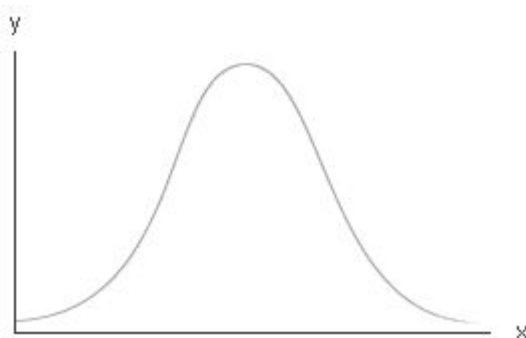


Figura 15: gráfico da curva normal

O resultado de z sempre indica a posição relativa da média da amostra em relação à média da população. Esse cálculo é normalizado, sendo que a média da população é o zero do gráfico, ou seja, o ponto de máximo da curva, o z representa a distância da amostra analisada em função da média da população, medidas em desvios padrões. Se temos como resultado da equação um $z = 2.23$, significa que a amostra analisada se encontra distante do zero 2.23 desvios padrões em relação à população. Para o modelo proposto, desejava-se que se uma mostra estivesse dentro dos limites da curva normal, existisse uma possibilidade de 95% de acerto de essa amostra ser realmente de uma pizza aceita pelos padrões de qualidade. Sendo assim, utilizando-se a tabela z , verifica-se que para a obtenção dessa probabilidade, o desvio padrão dever ser de no máximo 1,96, caso um z calculado seja maior que esse valor, existe 95% de chance de o mesmo não fazer parte da população, não estar dentro dos padrões de qualidades propostos.

Para o teste do sistema foram utilizadas quatro amostras de pizzas sendo que duas foram consideradas boas pelos entrevistados e as outras duas foram classificadas como não satisfatórias, sendo que uma delas estava assada demais e a outra estava pouco assada. Os valores de R, G e B dessas pizzas está disponível na tabela 2. A P1 e P2 foram classificadas como boas, a P3 como assada demais e a P4 como pouco assada. A figura 16 mostra as pizzas utilizadas para o teste do classificador.

	Queijo				Massa			
	P1	P2	P3	P4	P1	P2	P3	P4
R	188,82	208,36	236,19	167,94	166,61	137,84	194,93	102,83
G	160,76	148,53	214,44	95,415	127,17	79,291	145,23	73,455
B	88,044	73,915	144,02	57,380	68,489	40,26	80,213	54,652

Tabela 2: valores de R, G e B das pizzas utilizadas para teste

Utilizando a equação para o cálculo de z , e considerando que todas as seis variáveis devem estar dentro de seus respectivos intervalos, somente a pizza P2 foi considerada boa. A P1, que foi classificada como boa pela pesquisa teve todos os seus valores referentes ao queijo aprovado, porém, os seus valores da massa ficaram de fora. Os valores da massa ficaram todos acima do valor máximo permitido para cada valor. Analisando-se essa pizza com o

ColorInspector3D e fazendo uma análise visual, observou-se que a sua massa é mais clara do que a massa das pizzas utilizadas para se criar os valores de comparação. Isso se deve a forma como a mesma foi assada. A P1 foi assada durante 10 minutos, mas sob uma temperatura média de apenas 150°C em um forno diferente. Já a massa da pizza P4, que foi reprovada tanto na pesquisa quanto pelo teste z teve todos os seus valores da massa reprovados, mas só o valor de R da massa foi reprovado, tendo ficado do fora do intervalo por ser pequeno demais. A pizza P2 foi aprovada pelo sistema tendo todos os seus valores ficados dentro dos seus respectivos intervalos.



Figura 16: fotos das pizzas utilizadas para a verificação do sistema, em ordem de cima para baixo, da esquerda para a direita, P1, P2, P3 e P4.

Infelizmente o pequeno número de amostras para se determinar a média geral do sistema foi incapaz de fazer uma generalização maior. Isso se

mostrou bastante evidente principalmente com a variável B da massa. A mesma apresenta um desvio padrão muito baixo, possibilitando assim uma variação muito baixa nos valores de B da massa das pizzas. Acredito que com um número de amostras maior, esse valor tenderá a ser maior, fazendo assim com que o sistema tenha uma capacidade de generalização maior. Porém, em condições semelhantes de cocção, o sistema se mostrou capaz de realizar a classificação de uma pizza quanto ao seu grau de cocção.

4 Conclusão

Apesar do baixo número de amostras utilizado para a criação do padrão e das pizzas analisadas apresentarem condições diferentes de cocção, o sistema foi capaz de partindo da imagem de uma pizza, analisando e extraindo as características consideradas importantes para o processo, analisando estatisticamente os dados obtidos e tratando os mesmos de forma e retirar valores espúrios e comparando esses valores com uma população previamente analisada, foi possível classificar as pizzas utilizando uma abordagem estatística com uma taxa de acerto de 75%. O método possui limitações, pois, não foi analisado com outros sabores de pizzas, porém, como é possível se classificar utilizando uma ou duas características, existe uma grande possibilidade de o método funcionar para pizzas com outros tipos de cobertura. Devido a característica do método, é possível que o mesmo possa ser utilizado para outros tipo de alimentos, alimentos esses que possam ser analisados com uma foto de seu topo, como por exemplo, biscoitos, massas e muffins.

Limitações do método

Somente um tipo de pizza foi utilizado.

A quantidade de pizzas que foram utilizadas para se determinar o padrão foi baixa, fazendo assim com que o classificador tivesse uma capacidade de generalização baixa.

Metodologia para a obtenção das imagens deve ser melhorada, pois, houve interferência da luz ambiente.

Trabalhos Propostos

Comparação entre classificadores, comparar a abordagem neural contra a abordagem estatística.

Utilização de uma abordagem bayesiana para classificação de outros tipos de alimentos.

Automatização do processo de separação de características da imagem.

Análise de algoritmos para segmentação de imagem.

5 Referências

- Du, Cheng-Jin. and Sun, Da-Wen. (2005) "Comparison of three methods for classification of pizza topping using different colour space transformations", In: *Jornal of Food Engineering* 68, Elsevier.
- Sun, Da-Wen, Brosnan, Tadhg. (2003) "Pizza quality evaluation using computer vision – part 1 Pizza base and sauce spread", *Jornal of Food Engineering* 57, Elsevier.
- Sun, Da-Wen, Brosnan, Tadhg. (2003) "Pizza quality evaluation using computer vision – part 2 Pizza base and sauce spread", *Jornal of Food Engineering* 57, Elsevier.
- Francis, F. J. (1995) "Quality as influenced by color", *Food Quality and Preference* 6 (1995) p. 149-155, Elsevier.
- Yam, Kit L, Papadakis, E. S. (2004), "A simple digital imaging for measuring and analyzing color of food surfaces", *Jornal of Food Engineering* 61, Elsevier.
- Simões, A. S. and Costa, A. H. R. (2003). "Classificação de Laranjas Baseada em Padrões Visuais".
- Morgan, D. (2006) "Leeds Guide to OpenCV", <http://www.comp.leeds.ac.uk/vision/opencv/>, Junho de 2008.
- Ferreira, M., and Moraes, A. (2007) "Tutorial OpenCv - Tecgraf Puc-Rio", <http://www.tecgraf.puc-rio.br/~malf/opencv/index.htm>, Junho de 2008.
- Tenenbaum, J. B., Griffiths, T. L. and Kemp C, (2006), "Theory-based Bayesian models of inductive learning and reasoning", *TRENDS in Cognitive Sciences*, Vol. 10 N° 7 Julho 2006.
- Chater, N., Tenenbaum, J. B and Yuille, A., (2006) "Probabilistic models of cognition: Conceptual foundations", *TRENDS in Cognitive Sciences* Vol. 10 N° 7 Julho 2006.
- Wikipedia. (2008) "Minitab", <http://pt.wikipedia.org/wiki/Minitab>. 25 de Setembro de 2008.
- Wikipedia. (2008) "OpenCV", <http://pt.wikipedia.org/wiki/OpenCV>. 14 de Agosto de 2008.
- Deitel, H. M. and Deitel, P. J. "C++ Como Programar – Apresentando Projeto Orientado a Objeto com UML", Tradução de Carlos Arthur Lang Lisboa e Maria Lúcia Blanck Lisboa, Editora Bookman 3ª edição, 2004.
- Hoel, P. G., "Estatística Elementar", Tradução de Carlos Roberto Vieira Araújo, Editora Atlas, 2ª edição, 1979.
- Castro, L. S V. de, "Exercícios de Estatística", Editora Científica, 1970.
- Lapponi, J. C., "Estatística Usando Excel", 4ª Edição, Editora Campus, 2005.
- Witte, R. S., "Statistics", CBS College Publishing, 2ª Edição, 1985.

6 Anexos

Anexo A – Código Fonte

```
//=====
// Name      : GravaCoresArquivo.cpp
// Author    : Daniel de Aguiar
// Description : Programa que salva em um arquivo .txt os valore RGB e HSL de
//                               uma imagem
//=====
#include <string>
#include <sstream>

#include <cv.h>
#include <cvaux.h>
#include <highgui.h>
#include <stdio.h>

#include <math.h>

#include <iostream>

using std::cout;
using std::cin;
using std::ios;
using std::cerr;
using std::endl;

#include <fstream>

using std::ofstream;

#include <cstdlib>

#include <string>
#include <sstream>

using std::string;

void rgb2hsl( double, double, double, double &, double &, double & );
string substituiCaracter( string, const string, const string );
string double2String( double );

//void labelling2Frame::hsl2rgb( double, double, double, double, double, double
);

int main() {
    // Objeto da classe Scalar
    CvScalar s;
```



```

double R, G, B, H, S, L;

// construtor de ofstream abre o arquivo
ofstream saidaArquivo( "nome do arquivo de saída", ios::out );

// testa se abriu o arquivo
if( !saidaArquivo ) {
    cerr << "Arquivo não pode ser aberto" << endl;
    exit( 1 );
}

// img aponta para a imagem do endereço
IplImage* img = cvLoadImage(
"../imagens/filtros/pizza_03_final/pizza03_08_min_queijo.jpg" );

// grava os valores de R, G e B da figura no arquivo de saída
for ( int i=0; i < img->height; i++ ) {
    for ( int j=0; j < img->width; j++ ) {
        // pega os valores de RGB do pixel i,j da
        // imagem img
        s = cvGet2D( img, i, j );
        R = (double)s.val[ 2 ];
        G = (double)s.val[ 1 ];
        B = (double)s.val[ 0 ];
        // Retorna os valores HSL de uma figura RGB
        rgb2hsl( R, G, B, H, S, L );
        if ( R != 0 && G != 0 && B !=0 ) {
            // Converte um valor double para
            // string
            string h = double2String( H );
            // Substitui um caracter por outro
            h = substituiCaracter( h, ".", "," );
            string ss = double2String( S );
            ss = substituiCaracter( ss, ".", "," );
            string l = double2String( L );
            l = substituiCaracter( l, ".", "," );
            // imprime no arquivo os valores de i,
            // j, R, G, B, H, S e L
            saidaArquivo << j << " " << i << " "
            << R << " " << G << " "
            << B << " " << h << " " << ss << " " << l << "\n";
        }
    }
}
return 0;
}

```

// Retorna os valores HSL de uma imagem RGB

```

void rgb2hsl( double r, double g, double b, double &h, double &s, double &l ) {

    double var_R = ( r / 255.0 );    //Where RGB values = 0 - 255
    double var_G = ( g / 255.0 );
    double var_B = ( b / 255.0 );

    double var_Min = std::min( var_R, std::min(var_G, var_B) ); //Min.
value of RGB
    double var_Max = std::max( var_R, std::max(var_G, var_B) );
//Max. value of RGB
    double del_Max = var_Max - var_Min; //Delta RGB
value

    l = ( var_Max + var_Min ) / 2.0;

    if ( del_Max == 0 ) { //This is a gray, no chroma...
        h = 0; //HSL results = 0 - 1
        s = 0;
    } else { //Chromatic data...
        if ( l < 0.5 )
            s = del_Max / ( var_Max + var_Min );
        else
            s = del_Max / ( 2.0 - var_Max - var_Min );

        double del_R = ( ( var_Max - var_R ) / 6.0 ) + ( del_Max /
2.0 ) / del_Max;
        double del_G = ( ( var_Max - var_G ) / 6.0 ) + ( del_Max /
2.0 ) / del_Max;
        double del_B = ( ( var_Max - var_B ) / 6.0 ) + ( del_Max /
2.0 ) / del_Max;

        if ( var_R == var_Max )
            h = del_B - del_G;
        else if ( var_G == var_Max )
            h = ( 1.0 / 3.0 ) + del_R - del_B;
        else if ( var_B == var_Max )
            h = ( 2.0 / 3.0 ) + del_G - del_R;

        if ( h < 0 ) h += 1;
        if ( h > 1 ) h -= 1;
    } //end if no croma
}

// Converte um valor da classe String para Double

string double2String( double valor ) {
    std::ostringstream oss;
    oss << valor;
    std::string value = oss.str();
    return value;
}

```

```
}
```

```
// Substitui um caracter por outro em uma instancia da classe String
```

```
string substituiCaracter( string texto, const string velho, const string novo ) {  
    int x = texto.find( velho );  
    while ( x < string::npos ) {  
        texto.replace( x, 1, novo );  
        x = texto.find( velho, x + 1 );  
    }  
    return texto;  
}
```

```
}
```

```
/*
```

```
//-----
```

```
void labelling2Frame::hsl2rgb(double h,double s,double l,double &r,double  
&g,double &b){
```

```
    if ( s == 0 ) { //HSL values = 0 - 1  
        r = l * 255.0; //RGB results = 0 - 255  
        g = l * 255.0;  
        b = l * 255.0;
```

```
    } else {
```

```
        double var_2;
```

```
        if ( l < 0.5 )
```

```
            var_2 = l * ( 1 + s );
```

```
        else
```

```
            var_2 = ( l + s ) - ( s * l );
```

```
        double var_1 = 2.0 * l - var_2;
```

```
        r = round((255.0 * huetorgb( var_1, var_2, h + ( 1.0 / 3.0 )
```

```
)));
```

```
        g = round(255.0 * huetorgb( var_1, var_2, h ));
```

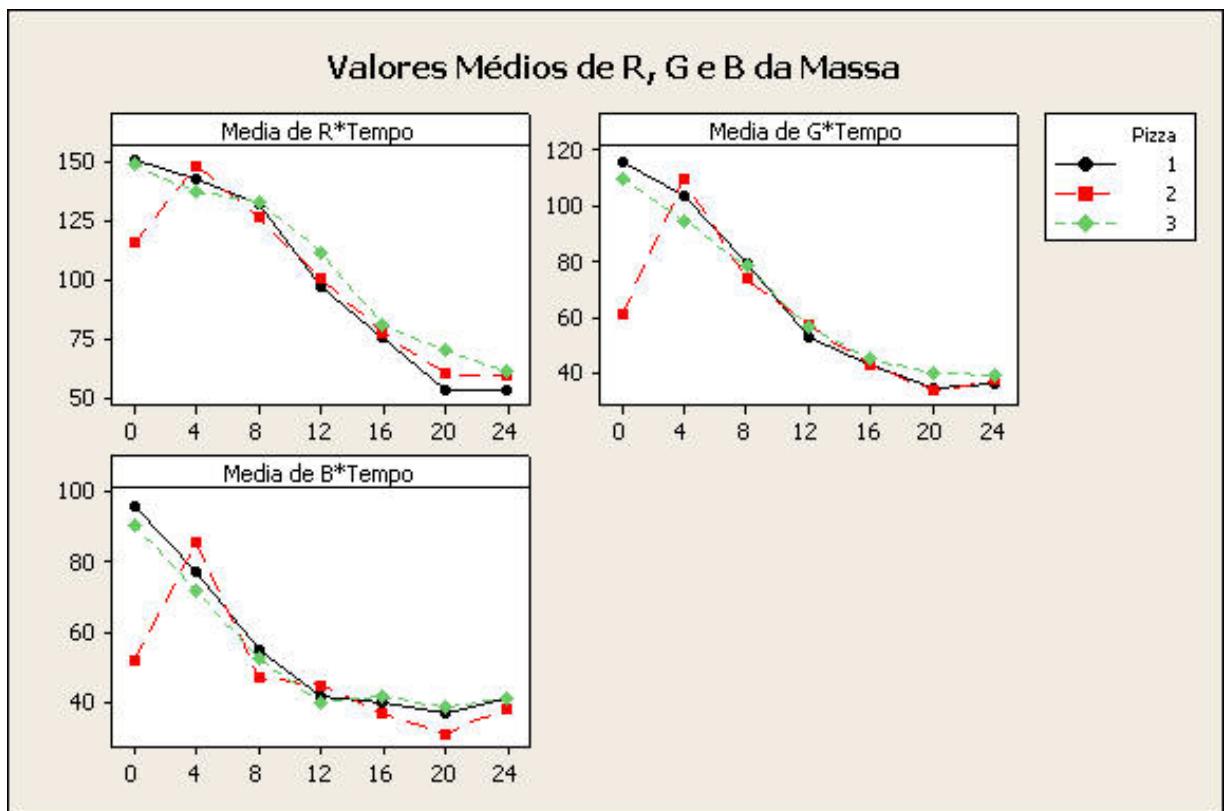
```
        b = round(255.0 * huetorgb( var_1, var_2, h - ( 1.0 / 3.0 )
```

```
));
```

```
    } //end if
```

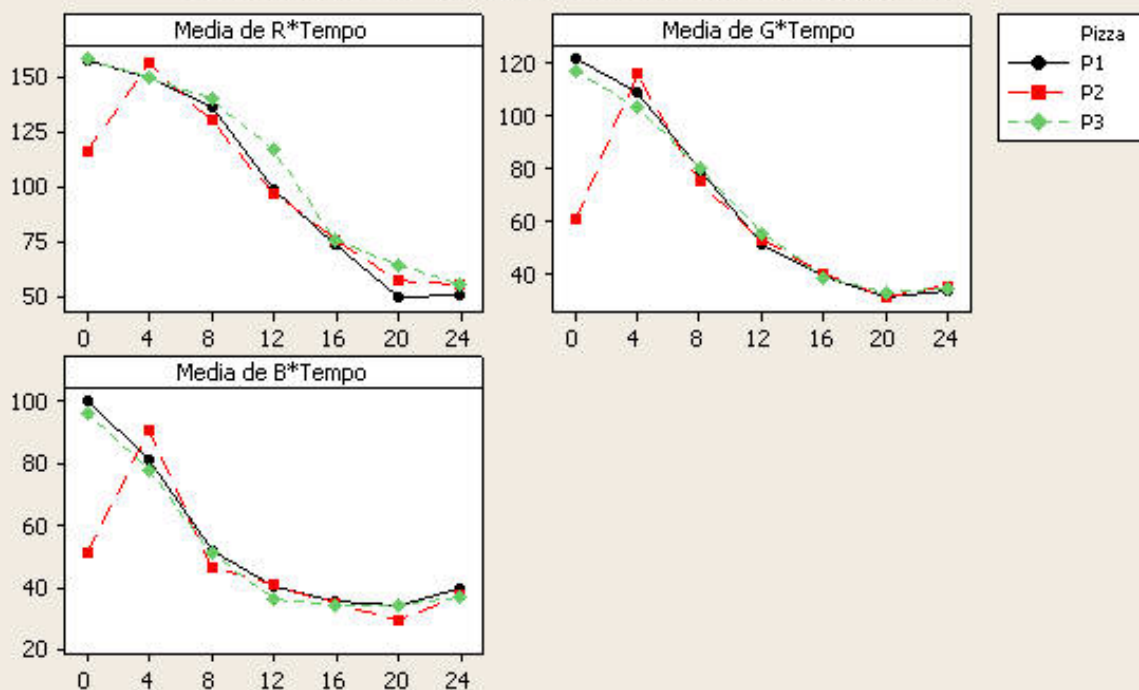
```
}//labelling2Frame::hsl2rgb */
```

Anexo B – Gráficos das Médias dos Valores



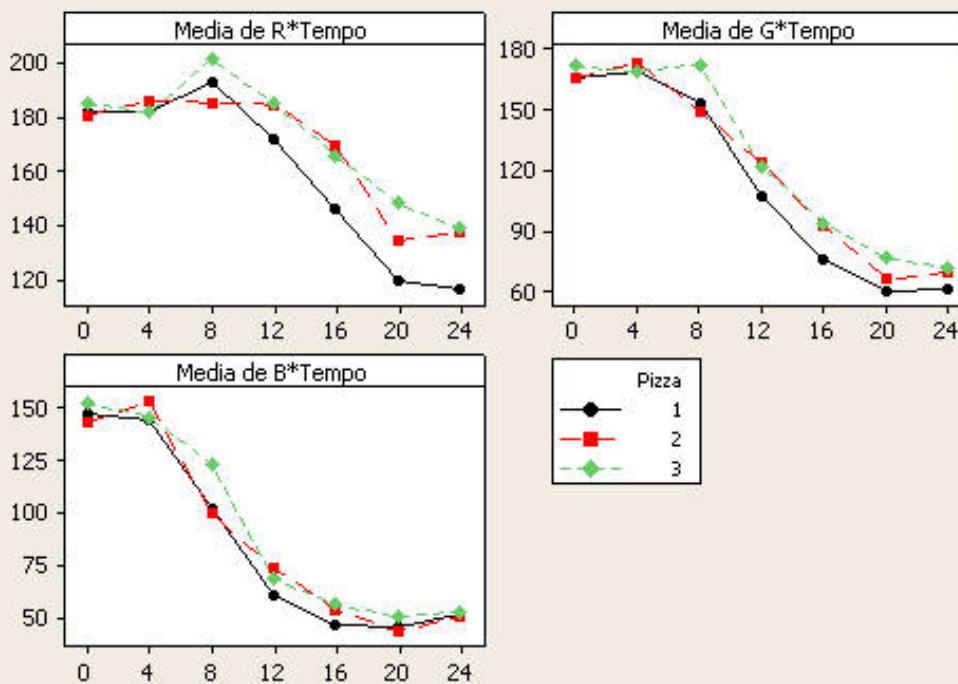
Valores médios da massa das 3 pizzas no espaço de cor RGB

Valores Médios de R, G e B da Massa após Quartil



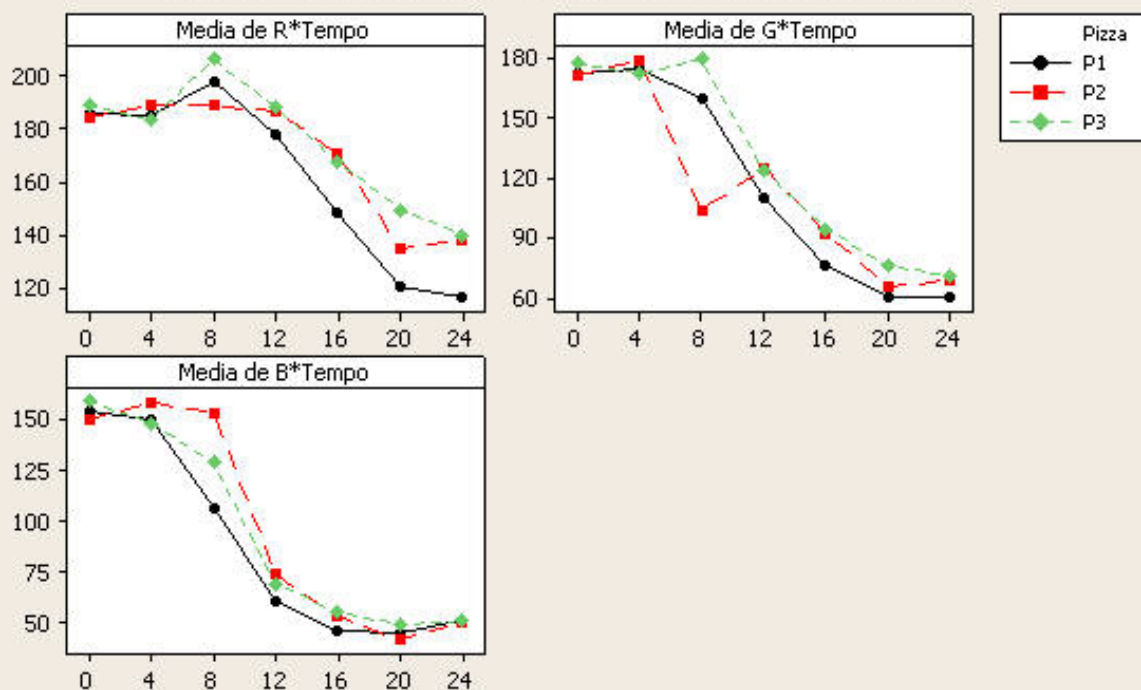
Valores médios da massa das 3 pizzas no espaço de cor RGB após retirada dos outliers

Valores Médios de R, G, B do Queijo



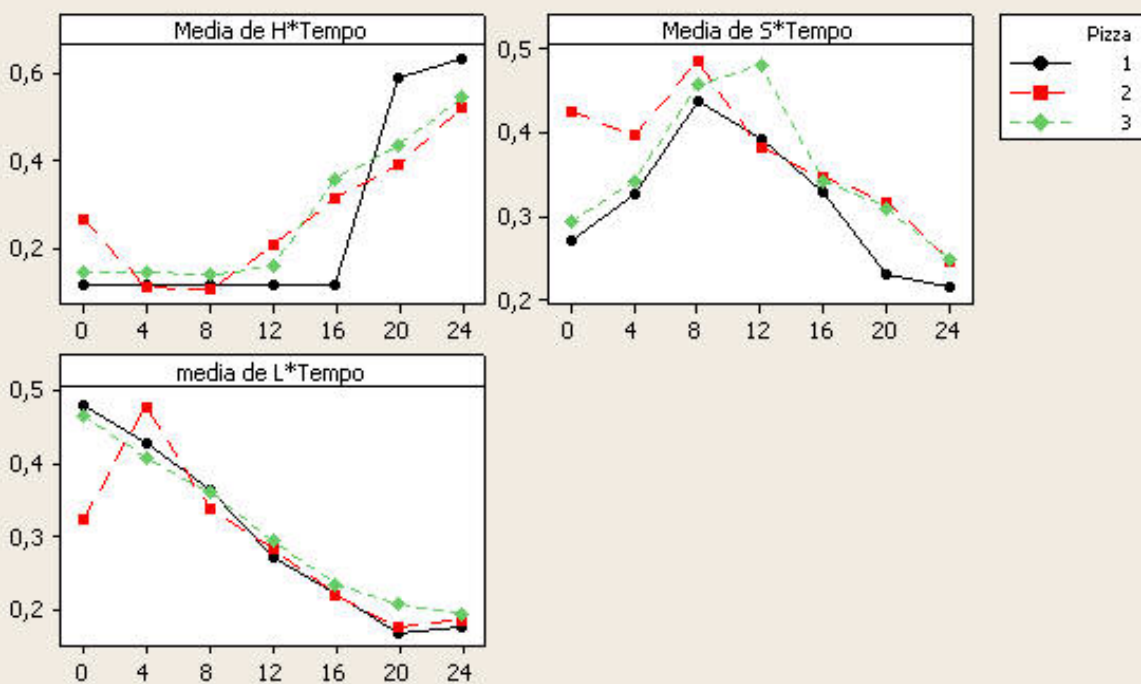
Valores médios do queijo das 3 pizzas no espaço de cor RGB

Valores Médios de R, G e B após o Cálculo dos quartis

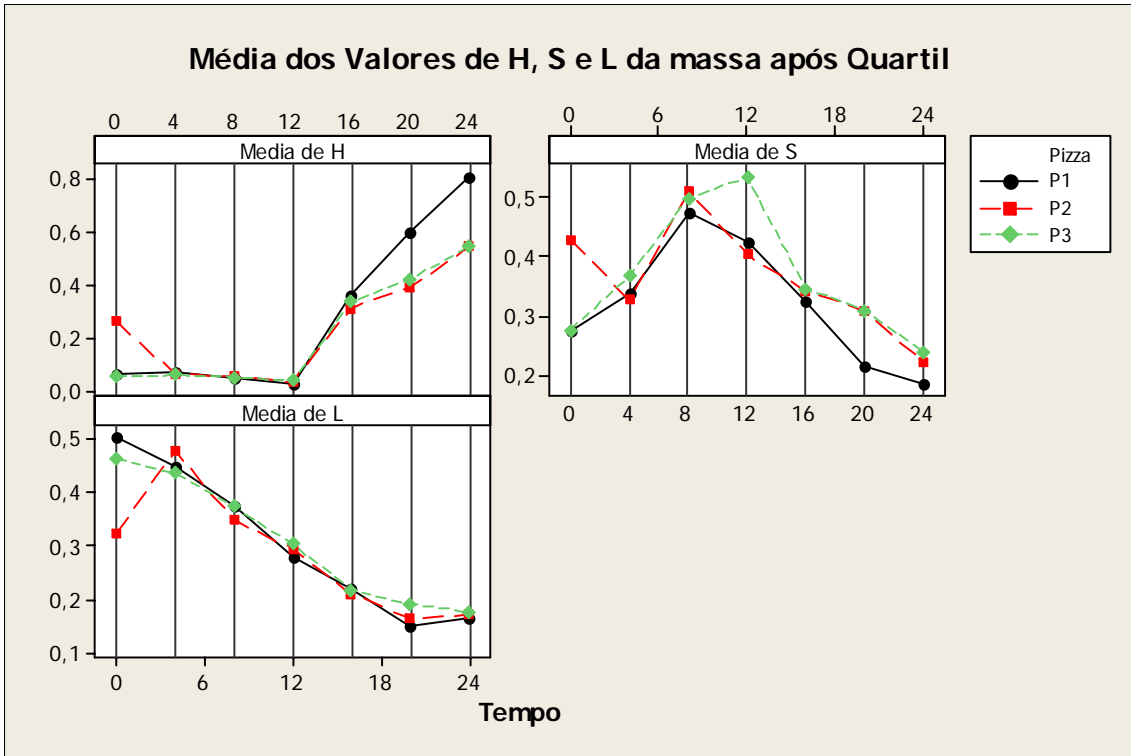


Valores médios do queijo das 3 pizzas no espaço de cor RGB após retirada dos outliers

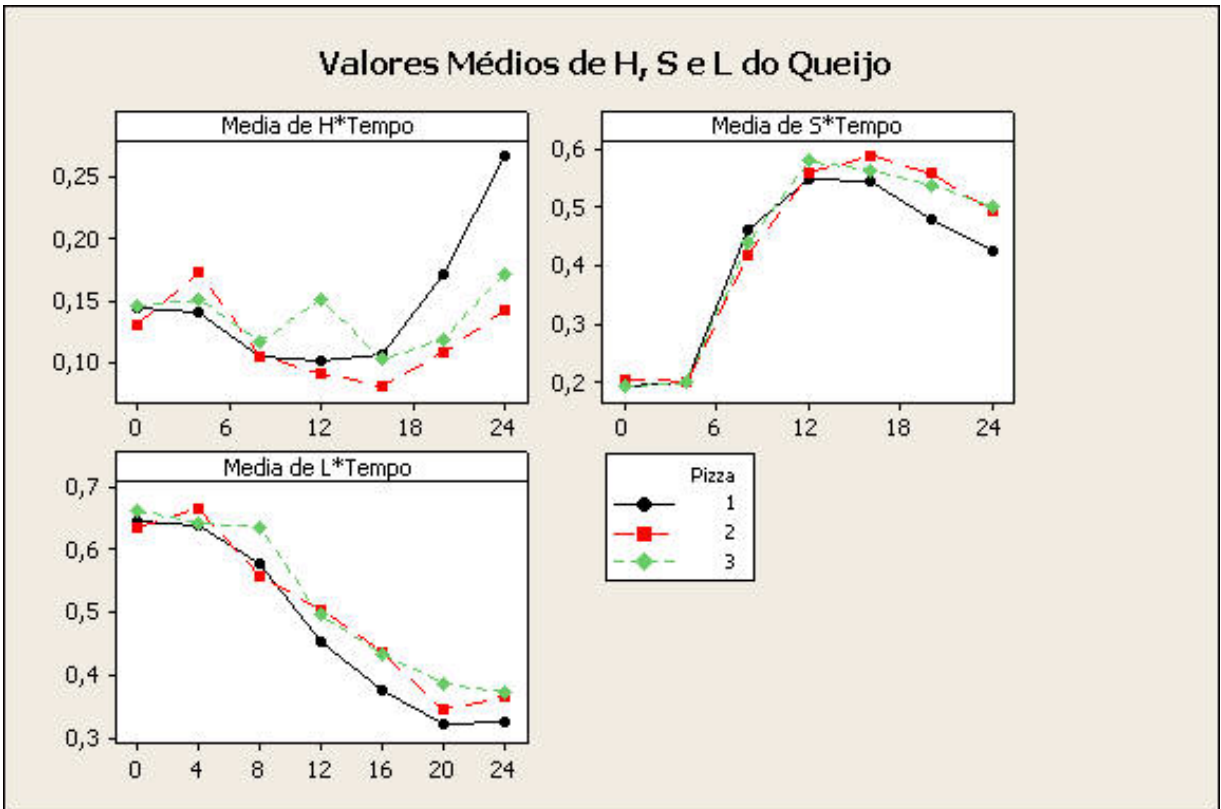
Valores Médios de H, S e L da Massa



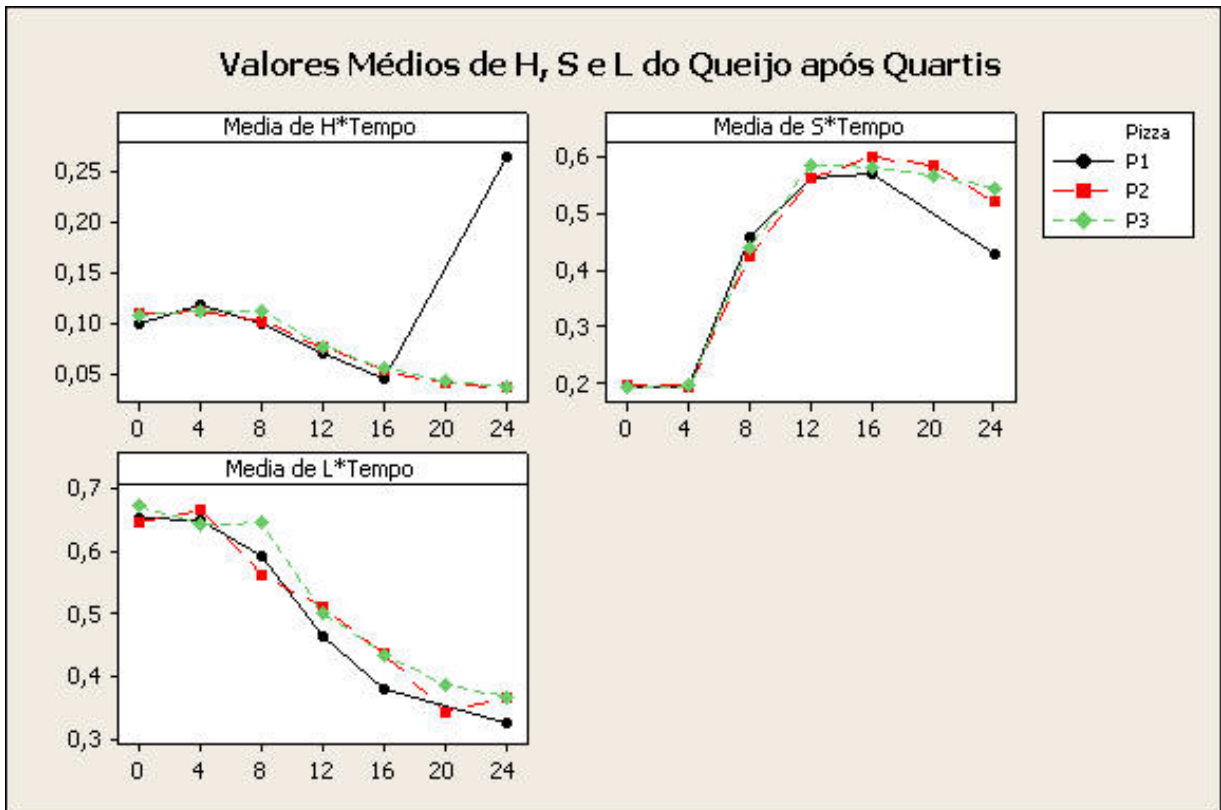
Valores médios da massa das 3 pizzas no espaço de cor HSL



Valores médios da massa das 3 pizzas no espaço de cor HSL após retirada dos outliers



Valores médios do queijo das 3 pizzas no espaço de cor HSL



Valores médios do queijo das 3 pizzas no espaço de cor HSL após retirada dos outliers

Anexo C – Código fonte para o cálculo dos quartis

```
#include <iostream>
```

```
using std::cout;
```

```
int main() {
    double qr1, qr3, qg1, qg3, qb1, qb3, qh1, qh3, qs1, qs3, ql1, ql3;
```

```
    // valores de R
```

```
    qr3 = "valor do 3º quartil de R";
```

```
    qr1 = "valor do 1º quartil de R";
```

```
    double iqdr = qr3 - qr1;
```

```
    double iqsr = qr3 + 1.5*iqdr;
```

```
    double iqir = qr1 - 1.5*iqdr;
```

```
    // valores de G
```

```
    qg3 = "valor do 3º quartil de G";
```

```
    qg1 = "valor do 1º quartil de G";
```

```
    double iqdg = qg3 - qg1;
```

```
    double iqsg = qg3 + 1.5*iqdg;
```

```
    double iqig = qg1 - 1.5*iqdg;
```

```
    // valores de B
```

```
    qb3 = "valor do 3º quartil de B";
```



```

qb1 = "valor do 1º quartil de B";
double iqdb = qb3 - qb1;
double iqsb = qb3 + 1.5*iqdb;
double iqib = qb1 - 1.5*iqdb;

// valores de H
qh3 = "valor do 3º quartil de H";
qh1 = "valor do 1º quartil de H";
double iqdh = qh3 - qh1;
double iqsh = qh3 + 1.5*iqdh;
double iqih = qh1 - 1.5*iqdh;

// valores de S
qs3 = "valor do 3º quartil de S";
qs1 = "valor do 1º quartil de S";
double iqds = qs3 - qs1;
double iqss = qs3 + 1.5*iqds;
double iqis = qs1 - 1.5*iqds;

// valores de L
ql3 = "valor do 3º quartil de L";
ql1 = "valor do 1º quartil de R";
double iqdl = ql3 - ql1;
double iqsl = ql3 + 1.5*iqdl;
double iqil = ql1 - 1.5*iqdl;

// imprime valores de retorno
cout << "IQSR: " << iqsr << "\n";
cout << "IQIR: " << iqir << "\n\n";

cout << "IQSG: " << iqsg << "\n";
cout << "IQIG: " << iqig << "\n\n";

cout << "IQSB: " << iqsb << "\n";
cout << "IQIB: " << iqib << "\n\n";

cout << "IQSH: " << iqsh << "\n";
cout << "IQIH: " << iqih << "\n\n";

cout << "IQSS: " << iqss << "\n";
cout << "IQIS: " << iqis << "\n\n";

cout << "IQSL: " << iqsl << "\n";
cout << "IQIL: " << iqil << "\n\n";

system("PAUSE");
return 0;
}

```

