

Criminal Plotter: Impressora CNC de esboço facial forense

Nícolas Auersvalt Marques * Laura Knoll Garcia de Oliveira **
Gabriel Lazari Trevisani ***

* Faculdade de Engenharia de Computação, Universidade Tecnológica Federal do Paraná, PR, (e-mail: nicolasauersvalt@alunos.utfpr.edu.br).

** Faculdade de Engenharia de Computação, Universidade Tecnológica Federal do Paraná, PR, (e-mail: lauraknoll@alunos.utfpr.edu.br)

*** Faculdade de Engenharia de Computação, Universidade Tecnológica Federal do Paraná, PR, (e-mail: trevisani@alunos.utfpr.edu.br)

Abstract: In 2023, Brazil recorded 47,722 murders and an average of 2,226 robberies and thefts per 100,000 inhabitants. It is even more concerning when considering that the crime resolution rate is 65

One of the factors responsible for the low case resolution is the shortage of resources and professionals: Brazil has an average of one criminal expert for every 41,500 inhabitants, while the UN recommends at least one for every 5,000. In the state of São Paulo, for example, experts travel up to 300 km in a single day to visit as many as 17 different crime scenes, meaning many incidents are overlooked, and in those that aren't, there's not enough time for a critical analysis. To reduce these dramatic numbers, the authors created a modified "CNC Plotter" printer that sketches criminal portraits of potential suspects based on a catalog of predefined facial features, eliminating the need for a forensic artist on-site.

Resumo: Em 2023, o Brasil registrou 47.722 assassinatos e uma média de roubos e furtos de 2.226 cada 100 mil habitantes. É mais preocupante quando se considera que a taxa de solução de crimes é 65%, sendo em torno de 35% quando se trata de homicídios.

Um dos fatores responsáveis pela baixa resolução de casos é a escassez de recursos e profissionais: o Brasil tem uma média de 1 perito criminal para cada 41.500 habitantes, enquanto a recomendação da ONU seria, no mínimo, 1 em cada 5.000. No estado de São Paulo, por exemplo, em um único dia, os especialistas chegam a percorrer 300 km em até 17 locais de crime diferentes, o que significa que muitas ocorrências são deixadas de lado e, nas que não são, não há tempo suficiente para uma análise crítica.

Para reduzir esses números dramáticos, os autores criaram uma impressora "CNC Plotter" modificada, que esboça retratos criminais de potenciais suspeitos com base em um catálogo de traços faceais pré-definidos, evitando a necessidade de ter um artista forense no local.

Keywords: Brazil; Crime resolution rate; Criminal expert; Shortage of resources; São Paulo; CNC Plotter; Criminal portraits; Forensic artist; Facial features.

Palavras-chaves: Brasil; Taxa de resolução de crimes; Perito criminal; Escassez de recursos; São Paulo; CNC Plotter; Retratos criminais; Artista forense; Traços faciais.

1. INTRODUÇÃO

Em 2023, o Brasil registrou 47.722 assassinatos em um ano, 10,4% do total mundial, segundo United Nations Office on Drugs and Crime (UNODC) (2023). Em homicídios per capita, está na 11ª posição, com 22,38 mortes a cada 100 mil habitantes, quase 1.000% da média da OCDE, 400% da média global e 150% da média na América Latina, de acordo com Organisation for Economic Co-operation and Development (2024). Ademais, em termos de roubos e furtos, o Brasil tem 2.226 para cada 100 mil habitantes, segundo o relatório de Centro de Ciência Aplicada à Segurança Pública da Fundação Getúlio Vargas (FGV CCAS) (2024).

A situação é mais preocupante quando se considera que a taxa de solução de crimes é 65%, consoante os estudos de Associação dos Delegados de Polícia do Brasil (ADEPOL) (2024), sendo em torno de 35% quando se trata de homicídios, segundo dados de Instituto Sou da Paz (2024).

Um dos fatores responsáveis pela baixa resolução de casos é a escassez de recursos e profissionais: segundo a reportagem da Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) (2024), o Brasil tem 1 perito criminal para cada 38.000 habitantes, enquanto a recomendação da Organização das Nações Unidas seria de, no mínimo, 1 para cada 5.000. No estado de São Paulo, por exemplo, em um único dia, os especialistas chegam a percorrer 300 km em até 17 locais de crime diferentes, o que significa que

muitas ocorrências são deixadas de lado e, nas que não são, não há tempo suficiente para uma análise crítica.

Os autores propõem, portanto, a criação de uma impressora "CNC Plotter"(adaptada) de esboço forense de rosto com fácil operação, baixo custo e baseada em métodos já existentes de retrato forense e reconhecimento facial, visando dar maior autonomia às testemunhas e reduzir a excedente demanda por peritos criminais. A autonomia das testemunhas é especialmente relevante em casos em que estas sofrem de Transtorno de Estresse Pós-Traumático (PTSD), visto que, conforme indicado por Morganstein et al. (2021) e Parra-Cardona and Wood (2021), indivíduos nessa condição podem apresentar dificuldades na comunicação de informações detalhadas.

2. REVISÃO DE LITERATURA

2.1 *O procedimento forense*

De acordo com Klum et al. (2013), esboços forenses são amplamente utilizados na ausência de registros fotográficos ou em vídeo do suspeito. Eles podem ser elaborados de duas formas: desenhados manualmente por um artista ou gerados digitalmente a partir da composição de elementos faciais armazenados em uma base de dados. O segundo método é mais recente e tem ganhado espaço, conforme McQuiston-Surrett et al. (2006).

Ainda de acordo com Klum et al. (2013), após a elaboração do esboço, a imagem pode ser divulgada em veículos midiáticos ou utilizada em buscas em bancos de dados criminais. Bahrum et al. (2023) apontam que essa busca pode considerar o rosto como um todo ou apenas algumas de suas características individuais. No presente trabalho, limitamo-nos exclusivamente à geração do esboço, sem aplicações diretas em reconhecimento facial automatizado.

2.2 *O método do esboço*

Geralmente preferível a biometria, DNA e outros métodos de identificação pela conveniência e custo relativamente menor, como explica Tang and Wang (2004), o esboço criminal forense ainda apresenta a inconveniência de precisar de peritos disponíveis com anos de prática nessa arte de acordo com Klum et al. (2013), realidade distante do contexto brasileiro. Uma alternativa, portanto, tem sido o esboço composto via software, que está sendo amplamente utilizado desde os anos 2000 segundo McQuiston-Surrett et al. (2006) e Galea and Farrugia (2016). No Brasil, no entanto, onde apenas recentemente foi inaugurado um sistema de reconhecimento facial, como notificado pelo Governo do Estado de São Paulo (2024), esse tipo de tecnologia ainda está longe de ser aplicada em larga escala, fazendo-se necessária uma adaptação da técnica.

2.3 *Identificação facial*

A teoria de identificação facial tem sua gênese na década de 1960, com os estudos de Bledsoe (1964) que utilizavam computadores e 20 medidas de componentes faciais inseridas pelo usuário. Na década de 1970, com os estudos de Harmon (1973) e Goldstein et al. (1971), foi proposto um algoritmo de busca binária para encontrar um rosto baseado em 21 características relacionadas aos componentes

	1	2	3	4	5
HAIR COVERAGE	FULL		RECEDING		BALD
LENGTH	SHORT		AVERAGE		LONG
TEXTURE	STRAIGHT		WAVY		CURLY
SHADE	DARK	MEDIUM	LIGHT	GRAY	WHITE
FOREHEAD	RECEDING		VERTICAL		BULGING
EYEBROWS WEIGHT	THIN		MEDIUM		BUSHY
SEPARATION	SEPARATED		MEETING		
EYES OPENING	NARROW		MEDIUM		WIDE
SEPARATION	CLOSE		MEDIUM		WIDE
SHADE	LIGHT		MEDIUM		DARK
EARs LENGTH	SHORT		MEDIUM		LONG
PROTRUSION	SLIGHT		MEDIUM		LARGE
CHEEKS	SUNKEN		AVERAGE		FULL
NOSE LENGTH	SHORT		MEDIUM		LONG
TIP	UPWARD		HORIZONTAL		DOWNTURNED
PROFILE	CONCAVE		STRAIGHT		HOOKED
MOUTH LIP THICKNESS (UPPER)	THIN		MEDIUM		THICK
LIP THICKNESS (LOWER)	THIN		MEDIUM		THICK
LIP OVERLAP	UPPER	NEITHER	LOWER		
WIDTH	SMALL		MEDIUM		LARGE
CHIN PROFILE	RECEDING		STRAIGHT		JUTTING
	1	2	3	4	5

Figura 1. Componentes faciais de Harmon

Fonte: Harmon (1973)

faciais, que se encontram na Figura 1. Baseado em uma descrição prévia de um rosto, os participantes eram dados 255 retratos e separavam em uma pilha os que continham a característica informada dos que não a tinham. Os resultados "surpreendentemente bons", nas palavras de Harmon, foram 53% de acerto com uma média de 7.3 características usadas, sendo que, em 75% dos casos, o retrato correto estava entre os últimos 13.

Segundo Samal and Iyengar (1992) e Adjabi et al. (2020), na década de 1980 esse campo de estudo ficou relativamente dormente, retornando na seguinte com muita intensidade, devido a diversos avanços tecnológicos, como

a inteligência artificial, que eliminou a necessidade de ter características subjetivas, como as anteriores. Nesse sentido, Turk and Pentland (1991), dois pesquisadores do MIT, inventaram o primeiro sistema de reconhecimento facial, "Eigenfaces". Em 1998, a agência governamental estadunidense DARPA criou a primeira base de imagens robusta, com 2.400 fotos de 850 pessoas.

Por fim, Adjabi et al. (2020) conclui que outra revolução ocorreu com o desenvolvimento do "Deep Learning" e redes neurais na década de 2010. Em 2014, por exemplo, a empresa "Facebook" afirmou que conseguia reconhecer faces com 97% da precisão humana com o algoritmo "Deepface", como exemplifica Taigman et al. (2014).

De modo geral, Hatem et al. (2015) Ding and Martinez (2010) explicam que os sistemas mais modernos variam entre uma abordagem de reconhecimento de componente versus reconhecimento da imagem como um todo. Em particular, na década de 2020, os modelos de "Deep Learning" evoluíram a ponto de estarem próximos da performance de um especialista em reconhecimento facial, por mais que a precisão máxima seja alcançada quando ambos trabalham em conjunto, como mostrado por O'Toole and Castillo (2021).

3. MATERIAIS E MÉTODOS

3.1 Catálogo de características faciais

Da revisão de literatura, entende-se que a abordagem mais adequada a realidade do país seria a criação de um catálogo que mimetiza o conceito de esboço composto sem a necessidade de um software. No entanto, se não há uma base de dados, surge o problema de como saber quantas características seriam necessárias para determinar o rosto aproximado do criminoso: sabe-se que, baseados nos modelos de Harmon, existem 9 componentes faciais divididas em 21 características que podem ser utilizadas para diferenciar rostos com segura assertividade, o que significa que elas podem ser utilizadas para gerar um banco de dados de componentes faciais, mimetizando o método de esboço via software.

Nesse sentido, os autores reduziram para 19 o número de características faciais mantendo o número de possibilidades de cada uma, expostas na Tabela 1, que foram utilizadas na criação das componentes do seguinte modo: os olhos, por exemplo, tem 3 características, cada uma com 3 possibilidades, o que implica em 27 tipos diferentes de olhos. O cabelo, por sua vez, conta com 19, pois "careca" não possui "tamanho" nem "textura". As bochechas, apenas 3, visto que só tem uma característica, e assim por diante. Assim, multiplicando as possibilidades de componentes, é possível montar aproximadamente 545 milhões de rostos distintos com um banco de dados de apenas 124 componentes.

A eliminação de "cor" de cabelo foi feita porque o plotter não consegue alterar a cor da impressão, enquanto a junção da característica de largura dos lábios, antes divididos em maior e menor, foi feita para que houvesse 27 tipos de boca no lugar de 81, que seria mais da metade dos desenhos de componentes.

Tabela 1. Tabela de características faciais de Harmon e Goldstein

Feature	1	2	3
Hair			
Coverage	Full	Receding	Bald
Length	Short	Average	Long
Texture	Straight	Wavy	Curly
Forehead	Receding	Vertical	Bulging
Eyebrows			
Weight	Thin	Medium	Bushy
Separation	Separated	Meeting	
Eyes			
Opening	Narrow	Medium	Wide
Separation	Close	Medium	Wide
Shade	Light	Medium	Dark
Ears			
Length	Short	Medium	Long
Protrusion	Slight	Medium	Large
Cheeks	Sunken	Average	Full
Nose			
Length	Short	Medium	Long
Tip	Upward	Horizontal	Downward
Profile	Concave	Straight	Hooked
Mouth			
Lip Thickness	Thin	Medium	Thick
Lip Overlap	Upper	Neither	Lower
Width	Small	Medium	Large
Chin Profile	Receding	Straight	Jutting

3.2 Impressora

Em consonância com o catálogo, os autores criaram uma impressora "CNC Plotter" modificada que, através da técnica de pontilhação, imprime o esboço com as componentes selecionadas pelo usuário. Como Samal and Iyengar (1992) demonstrou que a relevância das características faciais tende a reduzir do cabelo até o queixo, as componentes também são escolhidas nessa ordem.

3.3 CNC Plotter

O "CNC Plotter" é um projeto relativamente comum de eletrônica: utiliza-se um microcontrolador, um motor de passo e driver para cada haste da impressora, que pode ter 2 ou 3 dimensões (algumas versões utilizam um "servo motor" para o eixo Z), um adaptador "USB-Serial", caso necessário, um "CNC Shield" para conexão com o microcontrolador e o aparelho de desenho. Geralmente é programado com "G-code" (RS-274) e utilizado com concomitância com software de desenho de código aberto, como "Inkscape". Exemplos de projetos como o descrito são Girhe et al. (2018), Hasan et al. (2018), Nsayef and Mahmood (2018) e Hyder et al. (2020).

3.4 Materiais utilizados

- **Motor de passo 28BYJ-4 - 5V e driver ULN2003:** Como explica Hyder et al. (2020), um motor de passo é um motor elétrico DC sem escovas que divide uma rotação completa em um número de etapas iguais. O utilizado nesse projeto tem um ângulo de passo de $5,625^\circ$ com razão de redução de $\frac{1}{64}$. Isso significa que cada passo gira, aproximadamente, $0,088^\circ$, por mais que, como se pode ver na sessão 4.2, uma precisão

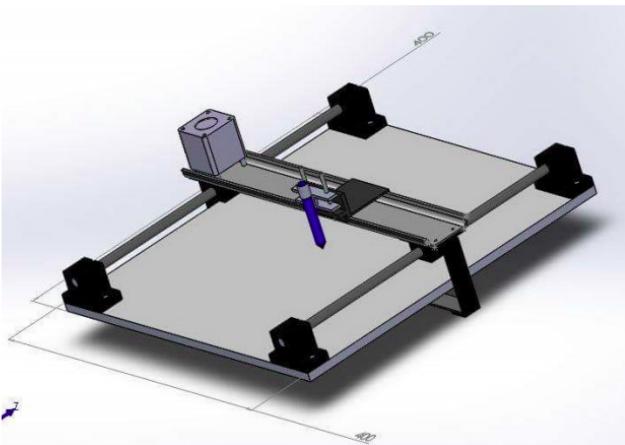


Figura 2. Exemplo de uma modelagem de um CNC Plotter
Fonte: Hasan et al. (2018)

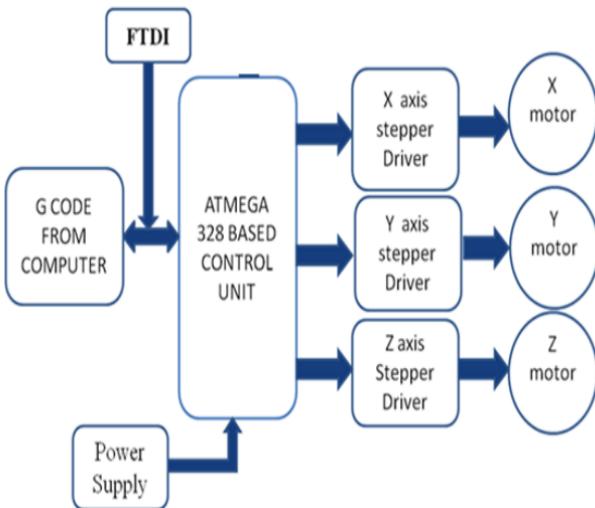


Figura 3. Exemplo de diagrama em bloco de um CNC Plotter
Fonte: Girhe et al. (2018)

assertiva não pode ser alcançada com um passo tão pequeno. Ademais, a função do motor de passo é controlar um eixo do "Plotter", e por isso foram utilizados 3 motores. Não se utilizou um motor Servo no eixo Z como Hyder et al. (2020) e Nsayef and Mahmood (2018) porque o projeto tem a limitação de não poder contar com uma "CNC Shield", e sim uma placa universal com solda. Também optou-se pelo driver ULN2003 em vez de uma Shield com L293D usada por Nsayef and Mahmood (2018), por ser mais flexível e garantir a utilização de portas digitais apenas para seus devidos motores. O motor de passo e seu respectivo driver se encontram na Figura 6.

- **Arduino Blackboard R3:** utilizou-se o Arduino Uno, um software e hardware de código aberto projetado e fabricado como uma única unidade de microcontrolador, como explica Hasan et al. (2018). Ele possui 14 portas digitais e 6 analógicas, e é o mais usado

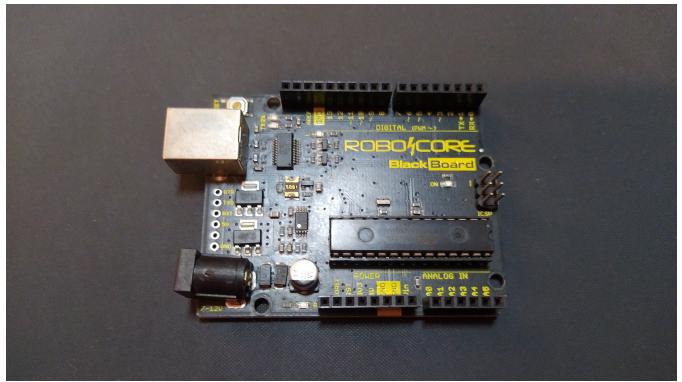


Figura 4. Arduino Uno utilizado
Fonte: autoria própria



Figura 5. Num Pad utilizado
Fonte: autoria própria

no "CNC Plotter" por sua fácil e intuitiva interface, Arduino IDE. No entanto, a limitação de memória do Arduino Uno teve que ser mitigada por um código python executado pelo computador, como explicado na sessão 3.6 e 3.7. Uma imagem do Arduino Uno utilizado se encontra na Figura 4.

- **Adafruit Membrane 3x4 Matrix Keypad:** Em decorrência da falta de portas do Arduino Uno, a linha 3, com os algarismos 7, 8 e 9, não pode ser utilizada. Isso adicionou outra limitação ao projeto, pois existem 9 componentes faciais e apenas 7 algarismos. Detalhes de como isso alterou o funcionamento se encontram na sessão 3.6.
- **Filamento PLA branco (500g):** A parte física do plotter foi modelada no Tinkercad (Figura 7 e Figura 8) e impressa sem custos no laboratório MEI-u da Universidade Tecnológica Federal do Paraná (UTFPR), campus Curitiba, utilizando um filamento PLA branco (500g). O modelo está ilustrado nas figuras 7 e 8, e os materiais impressos na figura 11.

O diagrama de blocos na figura 9 mostra como a estrutura se relaciona com os componentes, e o diagrama esquemático na figura 10 mostra a mesma relação em detalhes

3.5 Otimização de desempenho

Para garantir a padronização do ponto de início do desenho ((0,0)) e maximizar o alcance da área de impressão, optou-

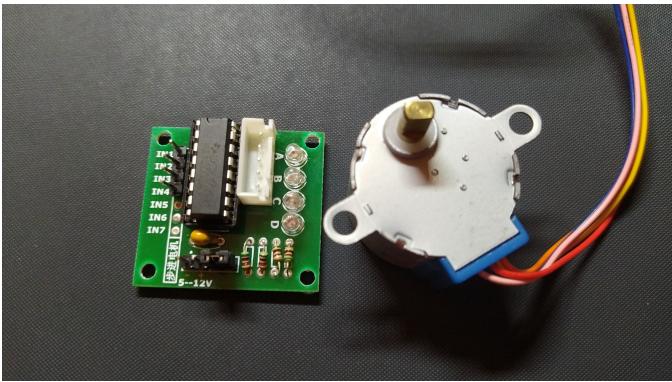


Figura 6. Motor de passo 28BYJ-4 - 5V e driver ULN2003
Fonte: autoria própria

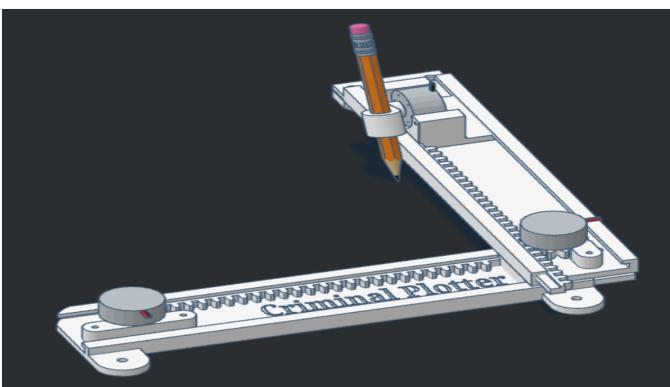


Figura 7. Modelagem do CNC plotter
Fonte: autoria própria

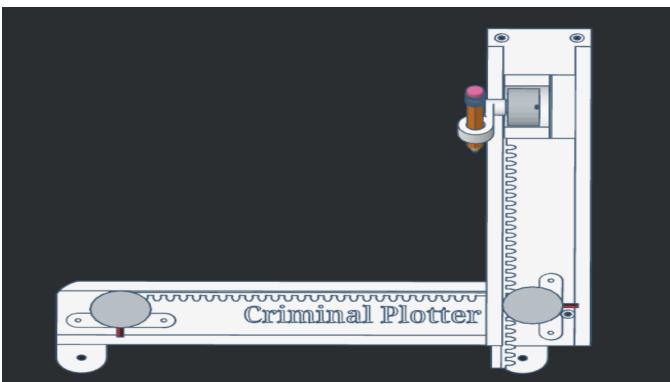


Figura 8. Modelagem do CNC plotter
Fonte: autoria própria

se por restringir a movimentação utilizando parafusos, sem a dependência de bibliotecas externas. O deslize contínuo sobre o PLA exigiu um processo intensivo de lixamento e testes, sendo que aplicou-se spray de silicone como lubrificante reduzir o atrito e garantir proteção extra as peças

Ademais, a movimentação da peça foi cuidadosamente ajustada para evitar deslocamentos indesejados durante o uso. A caneta foi posicionada em um ângulo fixo próximo a 45° em repouso, e, durante a descida, foi necessário evitar

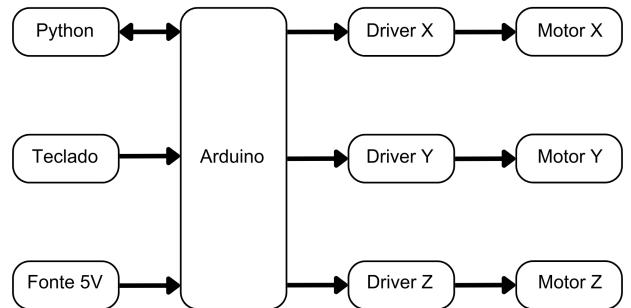


Figura 9. Diagrama em blocos do CNC Plotter
Fonte: autoria própria

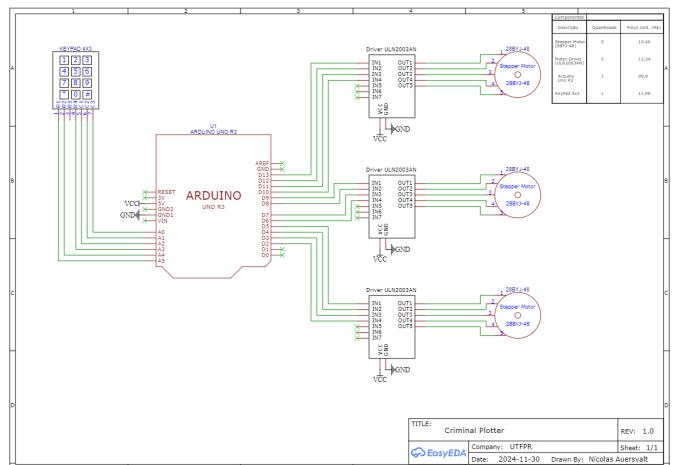


Figura 10. Diagrama esquemático do CNC Plotter
Fonte: autoria própria

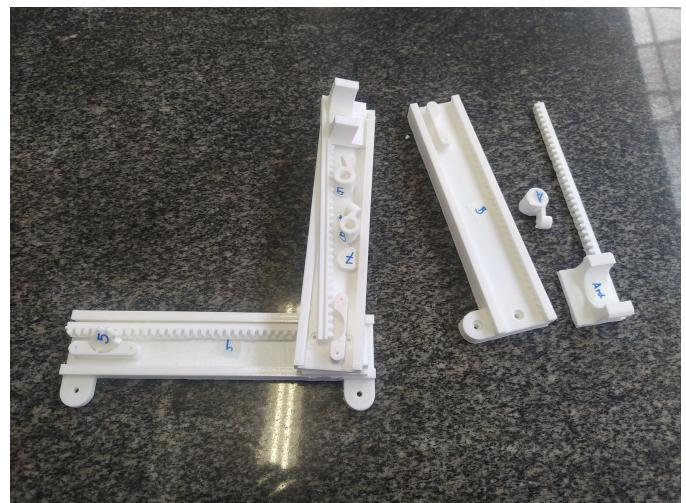


Figura 11. Material impresso com o filamento PLA
Fonte: autoria própria

que ela ficasse paralela ao eixo Y, pois essa configuração poderia causar riscos indesejados ou ultrapassar os limites da área de desenho, passando para um quadrante no qual não há contato com o papel. O mecanismo utiliza a própria limitação física da caneta tocando o papel para controlar sua movimentação, e a fixação da caneta foi realizada por pressão dentro de um compartimento específico, reforçada com parafusos.

3.6 O funcionamento do Criminal Plotter

O Criminal Plotter conta com uma série de códigos destinados à comunicação entre o computador e o Arduino, bem como à manipulação de imagens e matrizes. O processo, resumido na Figura 11, segue as etapas automatizadas abaixo para garantir eficiência e integração:

- (1) **Conversão de Imagens para Matrizes:** A execução inicia-se com um script que processa todas as imagens presentes no diretório `imagens`. Estas imagens, no formato 180x180 pixels em tons preto e branco (valores 0 ou 255), são convertidas em matrizes binarizadas. As matrizes resultantes são armazenadas no diretório `matrizes`, mantendo os nomes originais das imagens. Esses nomes servem como identificadores essenciais para as etapas seguintes.
- (2) **Comunicação Principal com o Arduino:** Na sequência, é executado o código principal em Python, responsável pela comunicação com o Arduino via serial, utilizando uma taxa de transmissão de 9600 baud. Esse código copia a matriz selecionada pelo usuário no diretório `matrizes` para o diretório “impressão”, depois lê a matriz correspondente e, por fim, transmite os dados ao Arduino para controle do dispositivo. A interação é contínua enquanto o código estiver em execução, permitindo que o usuário realize múltiplas impressões com diferentes matrizes.
- (3) **Soma de Matrizes no Diretório de Impressão:** Quando o usuário decide encerrar a impressão (pressionando `Ctrl+C`), um terceiro código em Python é acionado. Este realiza a soma de todas as matrizes presentes no diretório `impressão`, gerando uma matriz composta. A soma respeita uma limitação binária, não ultrapassando o valor 1 para garantir consistência nos resultados. Caso o usuário tenha omitido algum componente durante o processo de impressão, a matriz final poderá resultar em formas incompletas.
- (4) **Conversão Final e Limpeza de Matrizes:** Por fim, um último código converte a matriz composta de volta para o formato `PNG`, gerando uma imagem final que pode ser salva pelo usuário. Após a conversão, o diretório `impressão` é limpo automaticamente, apagando todas as matrizes para evitar a sobreposição de desenhos em futuras utilizações.

3.7 O código

O código desenvolvido em linguagem Arduino (`.ino`), cujo trecho está disponível na figura 12, diverge significativamente da literatura, dado a diferença radical de propósito. Ele inicia com a definição de variáveis globais amplamente utilizadas ao longo da execução, além do mapeamento do teclado Num Pad com suas limitações funcionais e a

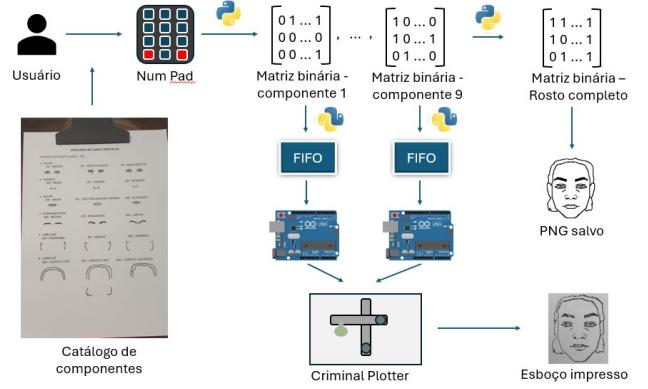


Figura 12. Diagrama do funcionamento do Criminal Plotter

Fonte: autoria própria

```

// Enquanto houver dados
while (Serial.available() >= 4)
{
    Serial.println("Pronto");

    byte x_bytes[2];
    byte y_bytes[2];

    // Lê os 2 bytes para x e 2 bytes para y
    x_bytes[0] = Serial.read();
    x_bytes[1] = Serial.read();
    y_bytes[0] = Serial.read();
    y_bytes[1] = Serial.read();

    // Converte os bytes para inteiros (considerando a ordem big-endian)
    int xPython = (x_bytes[0] << 8) | x_bytes[1]; // Concatena os dois bytes de x
    int yPython = (y_bytes[0] << 8) | y_bytes[1]; // Concatena os dois bytes de y

    // Direita (Antihorario)
    if (xAtual < xPython)
    {
        horario = false; // Direita
        mover(passo * (xPython - xAtual), motorX); // em X
    }
}
  
```

Figura 13. Trecho do código em linguagem Arduino (`.ino`)

Fonte: autoria própria

inicialização dos motores de passo. Após cada ação específica, é realizada uma sinalização, positiva ou negativa, que consiste na ativação sequencial e breve dos motores. Esse processo visa acionar os LEDs de cada módulo ULN2003, criando um feedback visual. No entanto, essa sinalização provoca pequenas rotações indesejadas nos motores, que precisam ser corrigidas posteriormente.

Em seguida, o programa entra no fluxo principal, que opera por meio de encadeamentos. A cada ciclo, o código aguarda dois inputs: o primeiro pode ser um valor numérico ou o caractere de cancelamento (utilizado para reiniciar os motores caso seja o primeiro ciclo), enquanto o segundo input deve ser obrigatoriamente um caractere de confirmação (*) ou de cancelamento (#). Esse processo de encadeamento é repetido três vezes, com diferentes sinalizações de eventos para indicar o progresso das operações.

Ao final do encadeamento, uma função chamada “desenhar” é executada. Essa função posiciona o motor Z na altura apropriada para a tarefa de desenho e envia um sinal “Pronto” para a interface Python, cujo trecho está

```

# Envia os pares da stack para o Arduino
while queue:
    x, y = queue.popleft()

    comando = f"({x}) ({y})\n"
    processed += 1
    progress = (processed / size) * 100

    print(f"Progresso: {progress:.2f}% concluído")

    # Exibe a mensagem de depuração
    print(f"Enviando coordenadas para o Arduino: [{comando}]")

    # Envia dois inteiros como bytes (2 bytes para cada inteiro)
    ser.write(x.to_bytes(2, 'big')) # Converte x para 2 bytes e envia
    ser.write(y.to_bytes(2, 'big')) # Converte y para 2 bytes e envia

    time.sleep(0.1) # Pequeno delay

    # Espera receber "OK" do Arduino (indica que terminou de imprimir)
    ok = 'L'
    while ok != 'Pronto':
        time.sleep(0.1) # Pequeno delay
        ok = ser.readline().decode('ascii', errors='ignore').strip() # Espera a linha completa de "OK"
        print(f"Recebido do Arduino: {ok}")

print("Impressão Concluída")

```

Figura 14. Trecho do código em linguagem Python

Fonte: autoria própria

disponível na figura 13, solicitando o envio de dois valores numéricos que correspondem às coordenadas x e y do próximo ponto. O código compara a posição atual com as coordenadas recebidas, determinando a direção de movimento necessária. Na posição alvo, é acionada a função “canetar”, responsável por baixar e levantar a caneta para marcar um ponto no local especificado. Após concluir a operação, o código retorna o sinal “Pronto” para o Python, indicando que está preparado para receber novos comandos.

Por fim, todo o fluxo é reiniciado em um loop contínuo, até que a comunicação com o Python seja interrompida.

O código Python, por sua vez, estabelece a comunicação com o Arduino de forma a garantir uma conexão estável entre os dispositivos. Inicialmente, ele aguarda o recebimento do sinal “Pronto” enviado pelo Arduino, acompanhado de três caracteres que identificam o nome de um arquivo. Com base nessa identificação (por exemplo, “123.txt”), o código busca no diretório de matrizes o arquivo correspondente.

Após localizar o arquivo, o código realiza a leitura da matriz nele contida e percorre seus elementos em busca de valores iguais a 1. Sempre que um valor 1 é encontrado, suas coordenadas são armazenadas em uma fila.

Ao término da leitura e processamento da matriz, o código entra em um laço onde cada posição armazenada na fila final é enviada ao Arduino, respeitando a sequência calculada previamente. Após cada envio, o código aguarda o sinal “Pronto” do Arduino antes de continuar com a próxima posição.

Por fim, o ciclo completo é repetido de forma contínua, garantindo o fluxo de comunicação e processamento entre o Python e o Arduino.

3.8 Custos

Atendendo ao propósito de ser um projeto de baixo custo, os gastos referentes ao “Criminal Plotter” estão detalhados na Tabela 2.

Componente	Gasto
Placa Arduino Uno + cabo USB	R\$56,00
Num Pad	R\$11,98
Motor de Passo + Driver ULN (x3)	R\$68,40
Jumpers F/M	R\$11,90
Placa universal	R\$8,00
Filamento PLA 500g (branco)	R\$75,00
Total	R\$231,28

Tabela 2. Custos do ”Criminal Plotter”

Fonte: autoria própria

É possível reconhecer?	Votos	% de votos
Sim	23	25
Parcialmente	46	51
Não	22	24
Total	91	100

Tabela 3. Votação realizada para os esboços extraordinários

Fonte: autoria própria

4. RESULTADOS E DISCUSSÕES

4.1 A qualidade do esboço e atuais limitações do catálogo

Dado a limitação de tempo na execução do projeto, foi possível imprimir apenas 2 rostos utilizando o catálogo. Eles estão nas figuras 15 e 16, respectivamente. Até o momento, não foi adicionado pelo facial nem acessórios, como brincos, óculos ou piercings, o que compromete significativamente a capacidade de relacionamento entre o esboço e pessoas reais. Nesse sentido, para que fosse testado apenas a qualidade do esboço, foi feitos 2 desenhos extraordinários, isso é, sem base no catálogo. Foi mostrado a imagem da pessoa, do png extraordinário e do esboço produzido a 90 participantes e perguntado se os esboços poderiam ser usado fielmente para reconhecer a foto dos dois participantes. Os resultados estão expostos na tabela 3 e a montagem com as fotos, desenhos e esboços se encontra na figura 17.

Assim, para que se teste o potencial do catálogo, ainda é necessário um modo científico de incluir os dois itens supracitados, algo que pode ser realizado em parceria com alguma delegacia de polícia especializada, como a Unidade de Execução Técnico-Científica de Curitiba - Centro - Polícia Científica, próxima da UTFPR. Essa mesma delegacia especializada pode auxiliar também a interpretar os resultados obtidos com a pesquisa e até mesmo melhorá-los.

Em relação ao código, a impressão foi otimizada em questão de tempo graças ao armazenamento das coordenadas com “1” em uma fila, separando o ato de ler a matriz do ato de percorrer o papel. Além disso, ainda foi possível calcular o quanto do desenho já tinha sido impresso, coisa impossível de ser feita sem a divisão dos atos.

No entanto, por mais que o processo tenha sido otimizado, um rosto com definição de 180x180 demora cerca de uma hora para ser desenhado, o que significa que qualidades superiores, como 780x780 ou até mesmo 1080x1080 seriam impraticáveis em um contexto real. No entanto, com base nos dados recolhidos, não se imagina que será necessária uma definição maior do que a utilizada.



Figura 15. Desenho 1 PNG feito pela composição de componentes (direita) e o respectivo esboço facial produzido pelo "Criminal Plotter"(esquerda)
Fonte: autoria própria

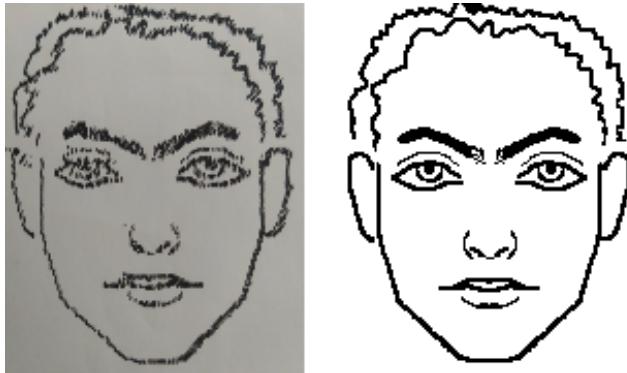


Figura 16. Desenho 2 PNG feito pela composição de componentes (direita) e o respectivo esboço facial produzido pelo "Criminal Plotter"(esquerda)
Fonte: autoria própria

4.2 Precisão e otimização

A precisão é proporcional a alguns fatores, como polimento físico, resolução das imagens e da matriz, posicionamento do papel (esticado), modelo da caneta e pressão aplicada pela mesma. Portanto, o Criminal Plotter não atua bem no desenho de matrizes com resolução menor que 100x100, e nem em impressão de palavras na ordem de 1-2 pixels de largura.

O pior tempo possível é o de desenhar no início e no final de cada linha, uma vez que precisa percorrer $O(n)$ para cada linha, e no pior caso sendo $O(n^2)$ desenhar um quadrado 180x180 preenchido. Ao terminar a inicialização preparatória para o desenho, deve-se desconectar o cabo do arduino e reconectá-lo para iniciar a impressão de um desenho, provavelmente por limitações do próprio arduino.

Por fim, o mencionado lixamento gerou espaçamentos que influenciaram diretamente o funcionamento. Como consequência, passos inferiores a 7 unidades ($0,615^\circ$, aproximadamente) não são bem interpretados pelo sistema da impressora. Nesses casos, o motor movimenta a engrenagem, mas o vão entre as peças impede a transmissão eficiente do movimento para o conjunto mecânico.

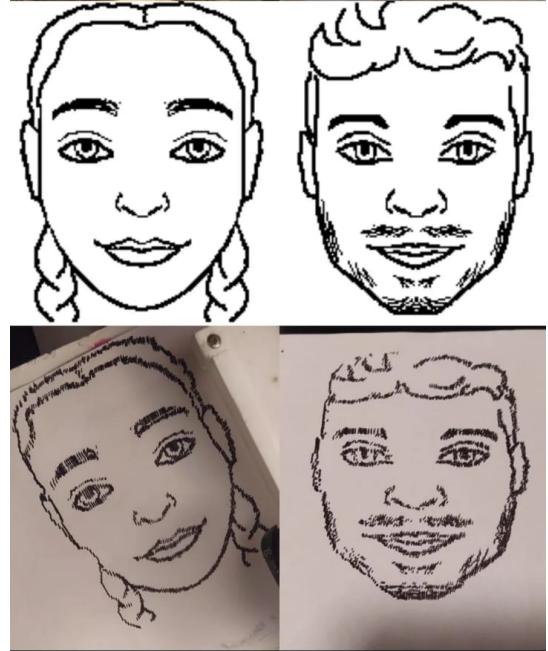


Figura 17. Montagem com foto, desenho png e esboço facial gerado pelo "Criminal Plotter" usado na pesquisa
Fonte: autoria própria

4.3 Dificuldades encontradas

- (1) Material usado na impressão 3D e a própria impressão exigiu acabamento completo de todas as peças.
- (2) Bibliotecas para os motores de passo (modelo-motor) não funcionaram corretamente, sendo testadas AccelStepper (Antiga e nova, ambas com problemas no controle de velocidade, desligamento dos motores, gerenciamento em cadeia e passos), Stepper (não funcionava a rotação no sentido contrário, desligamento dos motores e operava a velocidade muito baixa), FastAccelStepper (mesmos problemas da AccelStepper). Com isso, foi-se necessário controlar sem o uso de bibliotecas, mas sim atuando sobre cada polo dos motores a fim de girar nos dois sentidos. Além disso, necessitou-se descobrir a velocidade ideal para rotação, delay adequado e como desativar os motores para evitar consumo quando não usados. Um delay muito pequeno ou velocidade muito alta exigiam mais tensão dos apenas 5V disponíveis, fazendo com que os motores apenas vibrassem, sem giro.
- (3) As portas 0 e 1 do arduino são adequadas para comunicação RX (receive) e TX (transmit), impossibilitando o uso para o controle seja dos motores, como do keypad. Por causa disso e devido à limitação dos componentes, foi-se necessário desativar a última

- linha do keypad, pois não há portas suficientes no Arduino Uno R3.
- (4) Para conseguir a precisão necessária, necessitou-se alterações físicas no modelo a fim de limitar suas movimentações, haja vista que há muitas oscilações no circuito. Anteriormente, ao encostar e pressionar a caneta no papel, a parte física, mesmo estando parafusada corretamente, se movia e influenciava na precisão, devido à qualidade geral da impressão e lixamento. Para tanto, foi-se necessário fixar a caneta em um ângulo em que não ocorresse o levantamento do modelo. Além disso, alguns apoios (parafusos) foram usados para limitar tal problema.
 - (5) A alimentação externa havia sido uma alternativa, já que as bibliotecas não estavam conseguindo operar os motores de forma adequada. Entretanto, ao resolver o problema da biblioteca, descartou-se a necessidade de alimentação externa, uma vez que cada motor é operado por rotina.
 - (6) O arduino não é capaz de armazenar matrizes de ordem maior que 10, resultando em comportamentos inesperados. Para tanto, foi-se necessário comunicar-se com o computador através de uma biblioteca, em python, chamada pySerial, a qual se comunica com a Serial Port do arduino. Então as matrizes são lidas e processadas pelo python, e então enviadas as posições ao arduino.
 - (7) A soldagem da Shield foi outro fator de erro no projeto, por causa da qualidade do ferro de solda e de outros fatores como habilidades, pasta para solda usada e jumpers de baixa qualidade propiciou muitas oscilações nos componentes
 - (8) As repetidas correções de erros e otimizações do plotter foram tornando necessário um número mínimo de passos cada vez maior, alterando a resolução do projeto mais de 5 vezes. Isso atrasou significativamente a construção do catálogo, dado que os desenhos levam tempo para serem criados e a conversão dos pixels claramente não alcançou um nível de qualidade mínimo.

5. CONCLUSÃO

O projeto Criminal Plotter propõe uma solução acessível para a elaboração de esboços faciais em investigações criminais. A implementação de uma interface otimizada e de baixo custo busca democratizar o acesso às ferramentas de perícia no Brasil, contribuindo para a melhoria das investigações em um cenário de escassez de especialistas.

Apesar dos avanços obtidos, o tempo disponível para o desenvolvimento do projeto foi insuficiente para uma otimização completa do catálogo de traços faciais. Além disso, dificuldades técnicas relacionadas à precisão da impressão e à comunicação entre os dispositivos foram desafios significativos. Melhorias futuras devem focar na redução do tempo de impressão e na integração de motores de passo mais eficientes, como o Nema 17 usado por Hasan et al. (2018) e Hyder et al. (2020), o que exigiria adaptações estruturais na impressora.

Por fim, recomenda-se que futuros desenvolvimentos incluam colaboração com peritos criminais para aprimorar os parâmetros do catálogo de traços faciais, avaliar a qualidade dos esboços gerados e validar sua aplicabilidade em

casos reais. Essas melhorias podem fortalecer a contribuição do projeto para o avanço da investigação criminal no Brasil.

REFERÊNCIAS

- Adjabi, I., Ouahabi, A., Benzaoui, A., and Taleb-Ahmed, A. (2020). Past, present, and future of face recognition: A review. *Electronics*, 9(8), 1188.
- Associação dos Delegados de Polícia do Brasil (ADEPOL) (2024). Pesquisa de resolutividade dos inquéritos policiais na polícia civil dos estados e distrito federal e da polícia federal. Technical report, Associação dos Delegados de Polícia do Brasil, Brasília.
- Bahrum, N.N., Setumin, S., Abdullah, M.F., Maruzuki, M.I.F., and Che Ani, A.I. (2023). A systematic review of face sketch recognition system. *Journal of Electrical and Electronic Systems Research (JEESR)*, 22, 1–10.
- Bledsoe, W.W. (1964). The model method in facial recognition. Technical report, Panoramic Research, Inc., Palo Alto, CA, USA.
- Centro de Ciência Aplicada à Segurança Pública da Fundação Getulio Vargas (FGV CCAS) (2024). Evolução dos indicadores criminais no brasil.
- Ding, L. and Martinez, A.M. (2010). Features versus context: An approach for precise and detailed detection and delineation of faces and facial features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(11), 2022–2038.
- Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) (2024). A serviço da justiça. Acessado em: 18 out. 2024, disponível em aqui.
- Galea, C. and Farrugia, R.A. (2016). A large-scale software-generated face composite sketch database. In *2016 International Conference of the Biometrics Special Interest Group (BIOSIG)*, 1–5. IEEE.
- Girhe, P., Yenkar, S., and Chirde, A. (2018). Arduino based cost effective cnc plotter machine. *International Journal of Emerging Technologies in Engineering Research*, 6(2), 6–9.
- Goldstein, A.J., Harmon, L.D., and Lesk, A.B. (1971). Identification of human faces. *Proceedings of the IEEE*, 59(5), 748–760.
- Governo do Estado de São Paulo (2024). Governo inaugura laboratório de reconhecimento facial e digital da polícia civil. Acessado em: 18 out. 2024, disponível aqui.
- Harmon, L.D. (1973). The recognition of faces. *Scientific American*, 229(5), 70–83.
- Hasan, Y.M., Shakir, L.F., and Naji, H.H. (2018). Implementation and manufacturing of a 3-axes plotter machine by arduino and cnc shield. In *2018 International Conference on Engineering Technology and their Applications (IICETA)*, 25–29. IEEE.
- Hatem, H., Beiji, Z., and Majeed, R. (2015). A survey of feature base methods for human face detection. *International Journal of Control and Automation*, 8(5), 61–78.
- Hyder, S.F., Ibrahim, M., Adan, M.Z., and Mohammed, F. (2020). Cnc plotter machine. *Int. Res. J. Eng. Technol.*, 7(5), 5300–5305.
- Instituto Sou da Paz (2024). Onde mora a impunidade? Technical report, Instituto Sou da Paz, São Paulo.
- Klum, S., Han, H., Jain, A.K., and Klare, B. (2013). Sketch based face recognition: Forensic vs. composite sketches.

In *2013 international conference on biometrics (ICB)*, 1–8. IEEE.

McQuiston-Surrett, D., Topp, L.D., and Malpass, R.S. (2006). Use of facial composite systems in us law enforcement agencies. *Psychology, Crime & Law*, 12(5), 505–517.

Morganstein, J.C., Wynn, G.H., and West, J.C. (2021). Post-traumatic stress disorder: update on diagnosis and treatment. *BJPsych Advances*, 27(3), 184–186.

Nsayef, A.I. and Mahmood, A.L. (2018). Microcontroller-based plotter machine. *Al-Nahrain Journal for Engineering Sciences*, 21(3), 350–355.

Organisation for Economic Co-operation and Development (2024). Oecd better life index. URL <https://www.oecdbetterlifeindex.org/pt/quesitos/safety-pt/>.

O'Toole, A.J. and Castillo, C.D. (2021). Face recognition by humans and machines: three fundamental advances from deep learning. *Annual Review of Vision Science*, 7(1), 543–570.

Parra-Cardona, R. and Wood, L. (2021). Evaluation of technology-based advocacy services (eta): Technical report.

Samal, A. and Iyengar, P.A. (1992). Automatic recognition and analysis of human faces and facial expressions: A survey. *Pattern recognition*, 25(1), 65–77.

Taigman, Y., Yang, M., Ranzato, M., and Wolf, L. (2014). Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1701–1708.

Tang, X. and Wang, X. (2004). Face sketch recognition. *IEEE Transactions on Circuits and Systems for video Technology*, 14(1), 50–57.

Turk, M. and Pentland, A. (1991). Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1), 71–86. doi:10.1162/jocn.1991.3.1.71.

United Nations Office on Drugs and Crime (UNODC) (2023). Global study on homicide 2023. Technical report, United Nations, Vienna.