HYBRID AUTHENTICATION SYSTEM: INTEGRATING DEEPFAKE DETECTION AND FACIAL RECOGNITION

NURHANNAH BINTI MOHAMMAD KHAIRUL SHALEH

PROF DR SITI NORUL HUDA BT SHEIKH ABDULLAH

Fakulti Teknologi & Sains Maklumat, Universiti Kebangsaan Malaysia

ABSTRAK

Dalam era digital yang semakin berkembang, teknologi *deepfake* telah muncul sebagai ancaman serius terhadap keselamatan siber dan pengesahan identiti. Dengan menggunakan kecerdasan buatan (AI), teknologi ini berupaya menghasilkan imej, audio dan video yang sangat realistik, menjadikannya rentan terhadap penyalahgunaan dalam penyebaran maklumat palsu, penipuan identiti dan serangan siber. Kajian ini meneliti kelemahan model pengesanan *deepfake* sedia ada, khususnya model berasaskan Rangkaian Konvolusi Neural (CNN) yang cenderung menumpukan kepada ciri-ciri spatial sahaja tanpa mengambil kira kebergantungan temporal dalam urutan video. Kekurangan ini menyukarkan pengesanan manipulasi digital yang kompleks serta mengancam integriti sistem pengesahan biometrik. Bagi menangani isu ini, kajian mencadangkan model pengecaman *deepfake* berasaskan pendekatan spatiotemporal yang menggabungkan ResNet-50 untuk analisis dalam rangka dan *Bidirectional Long Short-Term Memory (*BiLSTM) untuk analisis urutan masa. Sistem ini turut digabungkan dengan pengecaman wajah bagi menambah lapisan keselamatan dalam aplikasi web. Model pengesahan identiti ini telah diuji secara menyeluruh dalam persekitaran simulasi bagi menilai tahap ketepatan dan kebolehpercayaannya sebelum dilaksanakan dalam aplikasi dunia sebenar. Sistem yang dibangunkan mencatat ketepatan sebanyak 96.13%, membuktikan potensinya dalam memperkukuh mekanisme pengesahan identiti serta membendung ancaman teknologi *deepfake* yang semakin canggih.

ABSTRACT

In the rapidly evolving digital era, deepfake technology has emerged as a serious threat to cybersecurity and identity authentication. Leveraging artificial intelligence (AI), this technology can generate highly realistic images, audio, and video content, making it vulnerable to misuse in the form of disinformation, identity fraud, and cyberattacks. This study investigates the limitations of existing deepfake detection models, particularly those based on Convolutional Neural Networks (CNN), which tend to focus solely on spatial features while overlooking temporal dependencies in video sequences. Such shortcomings make advanced deepfake content increasingly difficult to detect, posing significant risks to biometric authentication systems. To address these challenges, a deepfake recognition model is proposed based on a spatiotemporal approach that integrates ResNet-50 for intra-frame analysis and Bidirectional Long Short-Term Memory (BiLSTM) for temporal sequence learning. A hybrid identity authentication system was developed, combining deepfake detection with facial recognition to provide an additional layer of security in web-based applications. The system was rigorously tested in a simulation environment to evaluate its accuracy and reliability, serving as a critical phase prior to future deployment in real-world identity authentication applications. The proposed approach achieved an accuracy of 96.13%, demonstrating strong potential in enhancing identity verification mechanisms and mitigating the growing threats posed by deepfake technology.

PENGENALAN

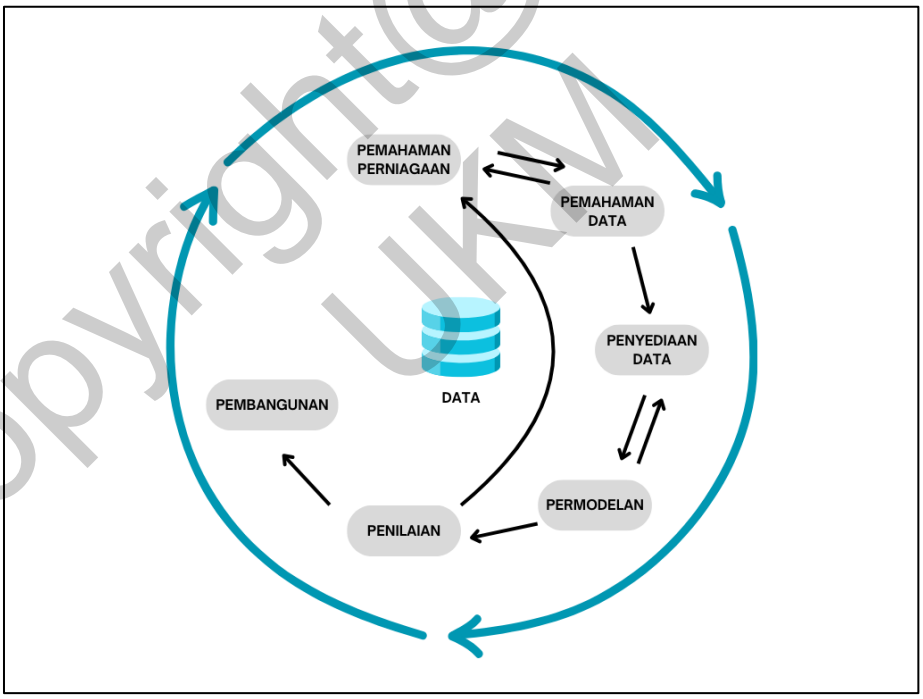
Dalam era digital yang semakin maju, teknologi *deepfake* telah muncul sebagai satu cabaran besar dalam bidang keselamatan siber dan pengesahan identiti. Menggunakan algoritma kecerdasan buatan (AI) untuk mencipta imej, audio atau video yang kelihatan sangat realistik, teknologi *deepfake* mempunyai potensi besar untuk disalahgunakan dalam pelbagai konteks. Ini termasuk penyebaran maklumat palsu, penipuan identiti, dan serangan siber (Arya et al. 2024). Keupayaan untuk mencipta kandungan yang sukar dibezakan daripada yang asli telah menjadikan teknologi ini semakin sukar untuk dikenalpasti (Johri & Arora 2022).

Salah satu cabaran utama dalam mengesan *deepfake* ialah kebanyakan model pengesanan sedia ada yang bergantung pada Rangkaian Neural Konvolusi (CNN). Model ini hanya memberi tumpuan kepada ciri-ciri spatial dalam imej atau video tetapi sering mengabaikan kebergantungan temporal dalam urutan video. Kekangan ini menjadikan CNN kurang berkesan dalam mengesan *deepfake* yang menunjukkan corak pergerakan tidak sekata, yang membolehkan *deepfake* yang canggih kelihatan sangat realistik (John & Sherif 2022). Ini menyukarkan lagi usaha untuk membezakan antara data biometrik yang asli dan yang telah dimanipulasi.

Selain itu, teknologi *deepfake* juga menimbulkan ancaman kritikal kepada sistem pengesahan biometrik, terutamanya dalam aplikasi pengecaman wajah. Individu yang tidak sepatutnya boleh menggunakan *deepfake* untuk mencipta penipuan identiti yang sangat realistik, membolehkan mereka mengelakkan sistem pengecaman wajah walaupun dengan penggunaan pengesanan masa nyata (Garg & Gill 2023). Dengan alat teknologi seperti DeepFaceLab, sesiapa sahaja kini mampu mencipta video palsu yang sangat realistik, yang seterusnya meningkatkan risiko akses tanpa kebenaran kepada data sensitif (Garg & Gill 2023).

METODOLOGI KAJIAN

Kajian ini menggunakan metodologi CRISP-DM *(Cross Industry Standard Process for Data Mining)* yang digabungkan dengan pendekatan *Agile* sebagai model pembangunan perisian bagi sistem pengesahan *deepfake* dalam simulasi laman web CSAM. Model ini merangkumi beberapa fasa utama iaitu Pemahaman Perniagaan, Pemahaman Data, Penyediaan Data, Permodelan, Penilaian dan Pembangunan. Rajah 1.1 menggambarkan keseluruhan fasa yang dilalui dalam CRISP-DM. Setiap fasa ini dikendalikan secara iteratif berdasarkan prinsip pecutan *Agile*, bagi memastikan sistem kekal fleksibel, menerima maklum balas secara berterusan serta membolehkan penambahbaikan sepanjang kitaran hayat projek. Pendekatan ini juga memastikan pengurusan data dilaksanakan secara menyeluruh, sambil mengurangkan keperluan untuk mengulang semula langkah awal semasa proses penambahbaikan sistem.



Rajah 1.1 Model CRISP-DM sebagai model pembangunan perisian

Fasa 1 Pemahaman Perniagaan

Fasa ini bertujuan untuk memahami kepentingan dan keperluan pengesanan *deepfake*, serta bagaimana teknologi yang telah dibangunkan dapat memberikan penyelesaian yang lebih berkesan. Dalam dunia digital yang semakin maju, kewujudan video

deepfake yang dihasilkan menggunakan kecerdasan buatan (AI) telah menjadi cabaran besar dalam aspek keselamatan siber, kredibiliti media dan kepercayaan awam terhadap kandungan digital. *Deepfake* telah digunakan untuk penipuan identiti, penyebaran maklumat palsu dan manipulasi politik, menjadikannya ancaman serius dalam pelbagai industri.

Bagi menangani masalah ini, pelbagai pendekatan dalam pengesanan *deepfake* telah diperkenalkan termasuk model berasaskan CNN dan RNN. Walau bagaimanapun, kajian terdahulu menunjukkan bahawa model-model ini mempunyai had dalam mengesan ketidakkonsistenan pergerakan dalam video *deepfake*, terutamanya apabila model hanya menumpukan kepada ciri spatial tanpa mengambil kira perubahan temporal dalam urutan bingkai video. Oleh itu, fasa ini memberi tumpuan kepada keperluan membangunkan model yang lebih cekap dalam mengesan manipulasi deepfake melalui gabungan pendekatan transformer dan rangkaian neural berulang (RNN).

Projek ini telah membangunkan model pengesanan *deepfake* yang menggabungkan model pralatih ResNeXt-50 dan BiLSTM *(Bidirectional Long Short-Term Memory)* bagi meningkatkan keberkesanan pengesanan. ResNet-50 telah digunakan untuk menganalisis setiap bingkai yang diekstrak, manakala BiLSTM membantu dalam mengesan perubahan temporal dan ketidakkonsistenan pergerakan sepanjang urutan video. Gabungan ini telah berjaya mengatasi kelemahan model sedia ada dengan menyediakan mekanisme pengesanan yang lebih komprehensif dan berkesan.

Fasa 2 Pemahaman Data

Fasa pemahaman data ini bertujuan untuk mengumpul dan meneroka data yang berkaitan dengan video palsu iaitu *deepfake* dan video asli. Tiga set data utama telah digunakan dalam projek ini, iaitu DFDC *(Deepfake Detection Challenge), FaceForensics++* (FF) dan *Celeb-DF*. Setiap dataset mengandungi video asli dan video palsu serta dilengkapi dengan fail metadata dalam format CSV yang menyenaraikan nama fail dan label klasifikasi sama ada "*REAL*" atau "*FAKE*". DFDC merupakan set data berskala besar yang telah dibangunkan oleh *Facebook* AI dan mengandungi ribuan

video *deepfake* dan video asal dengan pelbagai individu, latar belakang serta ekspresi wajah untuk mencerminkan situasi dunia sebenar. *FaceForensics++* pula ialah set data forensik yang mengandungi lebih 1,000 video asal yang telah dimanipulasi menggunakan teknik seperti *Deepfakes*, *Face2Face*, *FaceSwap* dan *NeuralTextures*, serta menyediakan variasi kualiti video untuk analisis yang lebih realistik. Sementara itu, *Celeb-DF* ialah set data i beresolusi tinggi yang dibina menggunakan video wawancara selebriti dan bertujuan untuk membetulkan kelemahan visual yang terdapat dalam set data *deepfake* terdahulu seperti ekspresi wajah yang tidak semula jadi atau penyesuaian bibir yang tidak tepat. Kesemua set data ini telah digunakan dalam bentuk yang telah diproses bagi bahagian muka sahaja bagi mempercepatkan proses pengecaman. Set data turut disertakan dengan label daripada fail CSV global yang digunakan untuk proses latihan dan penilaian model pengecaman video *deepfake*.

Rajah 1.2.1 menunjukkan contoh kandungan fail CSV yang telah digunakan sebagai metadata global dalam projek ini. Fail ini menyenaraikan nama fail video bersama label klasifikasi iaitu sama ada *'REAL'* atau *'FAKE'*. Label tersebut telah digunakan sepanjang proses latihan dan penilaian model pengecaman, dan memainkan peranan penting dalam memastikan ketepatan klasifikasi data input yang dimasukkan ke dalam sistem.

000.mp4	REAL
23186 unique values	FAKE 85% REAL 15%
000_003.mp4	FAKE
001.mp4	REAL
001_870.mp4	FAKE
002.mp4	REAL
002_006.mp4	FAKE
003.mp4	REAL
003_000.mp4	FAKE
004.mp4	REAL
004_982.mp4	FAKE
005.mp4	REAL
005_010.mp4	FAKE
006.mp4	REAL
006_002.mp4	FAKE

Rajah 1.2.1 Fail CSV yang mengandungi video yang dilabel

Rajah 1.2.2 menunjukkan paparan struktur fail bagi dataset *FaceForensics++* yang mengandungi video manipulasi wajah. Semua video dalam set ini telah diproses terlebih dahulu untuk hanya memaparkan bahagian muka individu. Dataset ini telah digunakan kerana ia merangkumi pelbagai teknik manipulasi seperti *FaceSwap* dan *Deepfakes* yang membolehkan model mempelajari perbezaan antara wajah asli dan wajah yang telah dimanipulasi secara lebih mendalam.

Rajah 1.2.2 Contoh video set data FF++

Rajah 1.2.3 menunjukkan struktur direktori bagi dataset *Celeb*-DF, yang terdiri daripada dua kategori utama iaitu video wajah palsu dan video wajah sebenar selebriti. Set data ini menampilkan kualiti video tinggi serta ekspresi wajah semula jadi, menjadikannya lebih mencabar dan sesuai digunakan untuk menguji ketepatan model dalam situasi realistik.

Rajah 1.2.3 Contoh video set data Celeb-DF

Rajah 1.2.4 menunjukkan struktur fail bagi dataset DFDC (*Deepfake Detection Challenge*). Video dalam set data ini juga telah diproses untuk hanya memaparkan bahagian muka. Set data ini merupakan salah satu set data terbesar dan paling pelbagai dari segi individu, latar, dan pencahayaan. Ia digunakan dalam projek ini untuk melatih model bagi tujuan pengecaman wajah *deepfake* yang lebih umum dan berkesan.

Rajah 1.2.4 Contoh video set data DFDC

Fasa 3 Penyediaan Data

Proses penyediaan data merupakan langkah kritikal dalam pembangunan sistem pengesanan *deepfake*. Dalam kajian ini, pelbagai set data video telah digunakan bagi memastikan model yang dibangunkan dapat mengenal pasti kandungan palsu dan tulen dengan lebih tepat. Sumber data yang digunakan termasuklah set data DFDC, *FaceForensics++* dan *CelebDF*.

Kesemua video ini telah diproses terlebih dahulu dengan mengekstrak bahagian wajah sahaja bagi mengurangkan gangguan daripada latar belakang serta menumpukan perhatian kepada ciri-ciri wajah yang lebih signifikan untuk pengesanan *deepfake*. Label bagi setiap video disimpan di dalam fail CSV (Gobal_metadata.csv) yang mengandungi maklumat nama fail dan jenis label iaitu FAKE atau REAL. Bagi memastikan proses latihan dan penilaian model berjalan secara adil dan seimbang, data telah dibahagikan kepada dua subset menggunakan fungsi *train_test_split*, iaitu set latihan (80%) dan set penilaian (20%).

Untuk memastikan keseragaman input, satu fungsi *custom_collate* telah dibangunkan bagi menangani isu perbezaan saiz dan panjang urutan bingkai video. Fungsi ini menggunakan kaedah *padding* iaitu menambah nilai sifar ke atas kekurangan dimensi agar semua data mempunyai bentuk yang sama sebelum dihantar ke dalam rangkaian neural. Data ini kemudiannya dibalut ke dalam kelas set data tersuai iaitu *VideoDataset*, dan digunakan untuk membina *DataLoader* bagi proses latihan dan penilaian. Setiap data sampel yang dimuatkan mengandungi satu urutan bingkai video berserta label yang sepadan.

Rajah 1.3.1 menunjukkan kelas *VideoDataset* telah dibangunkan bagi mengendalikan pemprosesan data video secara berstruktur sebelum dimasukkan ke dalam model pembelajaran mendalam untuk tugas pengesanan *deepfake*. Kelas ini digunakan untuk memuatkan data video secara bersiri, membaca label yang berkaitan dengan setiap video serta mengekstrak bingkai daripada setiap video dengan kaedah yang konsisten mengikut panjang urutan yang ditetapkan.

```
# Define the dataset class
class VideoDataset(Dataset):
    def __init__(self, video_names, labels, sequence_length=60, transform=None):
        self.video_names = video_names
        self.labels = labels
        self.transform = transform
        self.sequence_length = sequence_length
```

Rajah 1.3.1 Fungi VideoDataset()

Konstruktor kelas (__init__) menerima parameter yang terdiri daripada nama-nama fail video, fail label yang mengandungi label bagi setiap video, panjang urutan bingkai serta fungsi transformasi yang akan digunakan untuk setiap bingkai. Parameter

panjang urutan ini membolehkan sistem memastikan bahawa setiap video diproses kepada bilangan bingkai yang seragam bagi memudahkan pemprosesan dalam rangkaian neural.

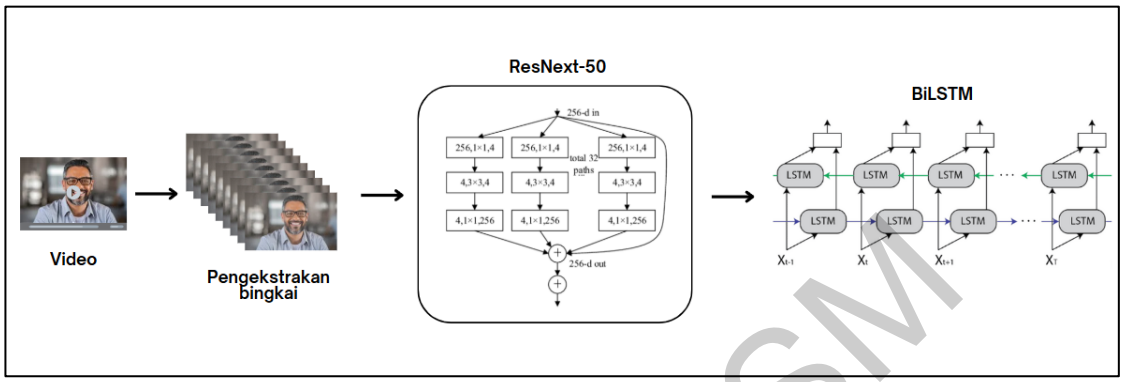
```python
def __getitem__(self, idx):
    video_path = self.video_names[idx]

    # Verify if label file exists and contains data
    if not os.path.exists(self.labels) or os.path.getsize(self.labels) == 0:
        raise FileNotFoundError("Labels file is empty or does not exist.")

    # Read labels data
    labels_df = pd.read_csv(self.labels, names=["file", "label"])

    # Handle empty labels or missing labels for a video
    try:
        label = labels_df.loc[labels_df['file'] == os.path.basename(video_path), 'label'].values[0]
        label = 0 if label == 'FAKE' else 1
    except IndexError:
        # Handle cases where no label is found for a video
        # You can choose to skip the video or assign a default label
        # Here, we'll raise an exception to indicate the issue
        raise ValueError(f"No label found for video: {video_path}")

    frames = []

    cap = cv2.VideoCapture(video_path)
    frame_count = int(cap.get(cv2.CAP_PROP_FRAME_COUNT))
    frame_step = max(frame_count // self.sequence_length, 1)
```

Rajah 1.3.2 Fungsi __getitem__()

Dalam kaedah *__getitem__* seperti yang ditunjukkan oleh Rajah 1.3.2, sistem akan membaca nama fail video berdasarkan indeks, memastikan fail label wujud dan mengandungi data. Jika tiada data dijumpai, pengecualian *FileNotFoundError* akan dijana. Kemudian fail CSV yang mengandungi nama fail video dan label *FAKE* atau *REAL* dibaca untuk mencari label yang sepadan dengan nama fail video yang sedang diproses. Label akan ditukar kepada format numerik (0 untuk FAKE dan 1 untuk REAL. Bingkai yang berjaya diproses akan disusun semula menggunakan *torch.stack* bagi membentuk *tensor* bersaiz (*sequence_length*, C, H, W) untuk dimasukkan ke dalam model.

Pendekatan ini memastikan setiap video diwakili oleh bilangan bingkai tetap dengan teknik praproses yang seragam serta menjamin label yang digunakan untuk latihan dan pengesahan adalah betul dan konsisten.

Fasa 4 Permodelan



Rajah 1.4.1 Cadangan seni bina algoritma model pengecaman *deepfake*

Bagi pembangunan sistem pengesanan *deepfake*, model pembelajaran mendalam telah dibangunkan dengan menggunakan gabungan rangkaian *convolutional* ResNeXt-50 dan BiLSTM bagi memproses data video yang telah diekstrak kepada bingkai bersaiz tetap seperti yang ditunjukkan pada Rajah 1.4.1. Rajah 1.4.2 menunjukkan potongan kod bagi pembangunan model. Model ini menggunakan pemberat ResNeXt-50 yang telah dilatih awal pada set data *ImageNet* untuk bertindak sebagai pengekstrakan ciri spatial setiap bingkai dalam video, di mana dua lapisan terakhir model ResNeXt-50 telah dibuang untuk mendapatkan peta ciri yang mengekalkan maklumat spatial penting. Peta ciri ini kemudiannya diproses menggunakan lapisan *Adaptive Average Pooling* bagi mengecilkan saiz spatial kepada bentuk vektor bersaiz 2048 bagi setiap bingkai video.

```python
# Define the model class
class Model(nn.Module):
    def __init__(self, num_classes, latent_dim=2048, lstm_layers=1, hidden_dim=2048, bidirectional=True):
        super(Model, self).__init__()
        model = models.resnext50_32x4d(pretrained=True)
        self.model = nn.Sequential(*list(model.children())[:-2])
        self.lstm = nn.LSTM(latent_dim, hidden_dim, lstm_layers, bidirectional=bidirectional)  # Set bidirectional=True
        self.dropout = nn.Dropout(0.4)
        self.linear = nn.Linear(hidden_dim * 2 if bidirectional else hidden_dim, num_classes)  # Adjust linear layer input size
        self.avgpool = nn.AdaptiveAvgPool2d(1)
```

Rajah 1.4.2 Kelas Model

Vektor-vektor ciri yang diperoleh daripada setiap bingkai akan disusun mengikut urutan masa sebelum dimasukkan ke dalam lapisan BiLSTM. Lapisan BiLSTM ini digunakan bagi menangkap hubungan temporal antara bingkai dalam sesuatu video, membolehkan model mengesan perbezaan pergerakan wajah atau ekspresi yang mungkin menunjukkan ciri-ciri *deepfake*. Hasil keluaran daripada

BiLSTM akan diringkaskan menggunakan purata merentasi urutan masa bagi membentuk satu vektor representasi bagi keseluruhan video. Vektor ini kemudiannya melalui lapisan *dropout* untuk mengurangkan risiko *overfitting* semasa latihan model. Seterusnya, hasil keluaran daripada *dropout* akan diproses melalui lapisan linear bersambung yang berfungsi sebagai lapisan klasifikasi akhir bagi menghasilkan keputusan sama ada video tersebut adalah *deepfake* atau video sebenar berdasarkan ciri-ciri yang telah dipelajari.

```
def forward(self, x):
    batch_size, seq_length, c, h, w = x.shape
    x = x.view(batch_size * seq_length, c, h, w)
    fmap = self.model(x)
    x = self.avgpool(fmap)
    x = x.view(batch_size, seq_length, -1)  # Adjust view to accommodate bidirectional LSTM output
    x_lstm, _ = self.lstm(x)
    return fmap, self.dropout(self.linear(torch.mean(x_lstm, dim=1)))
```

Rajah 1.4.3 Fungsi forward()

Dalam kaedah *forward*, bentuk input asal iaitu *(batch_size, sequence_length, channels, height, width*) diratakan *kepada (batch_size * sequence_length, channels, height, width)* bagi membolehkan pemprosesan oleh ResNeXt secara individu untuk setiap bingkai. Selepas proses pengekstrakan ciri dan pooling, input dibentuk semula sebagai urutan sebelum dihantar ke BiLSTM. Akhirnya, output akhir daripada LSTM digunakan untuk membuat klasifikasi video sama ada palsu atau tulen. Gabungan kedua-dua pendekatan spasial (ResNeXt) dan temporal (BiLSTM) ini memberikan kelebihan dalam konteks pengesanan *deepfake* kerana ia membolehkan model memahami corak perubahan wajah dalam satu jujukan video, bukan sekadar menilai bingkai secara berasingan.

Latihan dan Pengujian Model Pengecaman Deepfake

```
# Define transformations
im_size = 112
mean = [0.485, 0.456, 0.406]
std = [0.229, 0.224, 0.225]
train_transforms = transforms.Compose([
    transforms.ToPILImage(),
    transforms.Resize((im_size, im_size)),
    transforms.ToTensor(),
    transforms.Normalize(mean, std)
])
```

Rajah 1.4.4 Proses penukaran imej untuk latihan

Rajah 1.22 menunjukkan proses transformasi bagi memastikan data video dapat diproses secara seragam sebelum dimasukkan ke dalam model pembelajaran mendalam, transformasi pra-pemprosesan telah ditetapkan menggunakan *torchvision.transforms*. Transformasi ini digunakan untuk setiap bingkai video yang diekstrak daripada set data sebelum dihantar ke model semasa latihan. Transformasi ini telah dimasukkan ke dalam pemboleh ubah *train_transforms* yang akan digunakan semasa pemuatan data ke dalam *VideoDataset* bagi latihan model. Penggunaan transformasi ini membantu model dalam meningkatkan kestabilan dan keberkesanan semasa proses latihan kerana setiap bingkai akan diproses dengan konsisten dari segi saiz, format dan skala nilai pixel.

Setelah model pembelajaran mendalam dibangunkan, proses latihan dan pengujian dijalankan untuk mengoptimumkan parameter model serta menilai keupayaannya dalam mengesan video *deepfake*. Proses ini merangkumi penggunaan fungsi dan fungsi pengujian yang direka bentuk untuk mengendalikan satu *epoch* latihan dan penilaian prestasi terhadap set data penilaian.

```
def train_epoch(epoch, num_epochs, data_loader, model, criterion, optimizer):
    model.train()
    losses = []
    accuracies = []
    for i, (inputs, targets) in enumerate(data_loader):
        if torch.cuda.is_available():
            targets = targets.type(torch.cuda.LongTensor)
            inputs = inputs.cuda()
        _, outputs = model(inputs)
        loss = criterion(outputs, targets)
        acc = calculate_accuracy(outputs, targets)
        losses.append(loss.item())
        accuracies.append(acc)
        optimizer.zero_grad()
        loss.backward()
        optimizer.step()
        sys.stdout.write(
            "\r[Epoch %d/%d] [Batch %d / %d] [Loss: %f, Acc: %.2f%%]"
            % (epoch, num_epochs, i, len(data_loader), np.mean(losses), np.mean(accuracies)))
    torch.save(model.state_dict(), 'checkpoint.pt')
    return np.mean(losses), np.mean(accuracies)
```

Rajah 1.4.5 Fungsi train_epoch()

Semasa latihan, data video daripada set latihan dibahagikan kepada *batch* menggunakan *DataLoader* dan setiap *batch* dimasukkan ke dalam model. Model diaktifkan dalam mod latihan *(model.train())* dan setiap *batch* diproses untuk mendapatkan output ramalan. Output tersebut dibandingkan dengan label sebenar

menggunakan fungsi kehilangan *CrossEntropyLoss* yang sesuai digunakan dalam klasifikasi dua kelas. Fungsi kehilangan ini mengira perbezaan antara ramalan model dan jawapan sebenar dan nilai kehilangan ini kemudiannya digunakan untuk mengemas kini parameter model melalui kaedah *backpropagation*.

Pengoptimuman dilakukan menggunakan *Adam Optimizer* dengan kadar pembelajaran sebanyak 1e-5 serta kadar peluruhan berat yang sama. Dalam setiap iterasi, ketepatan juga dikira menggunakan fungsi *calculate_accuracy* yang mengira peratusan ramalan yang betul dalam sesuatu *batch*. Setelah semua *batch* bagi satu *epoch* selesai diproses, purata kehilangan dan ketepatan disimpan dan model disimpan ke cakera menggunakan *torch.save()*.

Bagi memantau prestasi model sepanjang proses latihan, dua plot penting dijana iaitu plot kehilangan (*loss*) dan ketepatan (*accuracy*) bagi data latihan dan data penilaian merentas semua *epoch*. Ini membolehkan pemantauan terhadap tingkah laku model, termasuk pengesanan awal terhadap isu seperti *overfitting* (jika ketepatan penilaian menurun manakala ketepatan latihan terus meningkat).

Secara keseluruhannya, proses latihan dan pengujian ini dilaksanakan sebanyak 20 *epoch*. Hasil daripada setiap *epoch* menunjukkan sama ada model semakin memahami pola data *deepfake* atau tidak. Model yang menunjukkan prestasi terbaik dalam ujian kemudiannya boleh dipilih sebagai model akhir untuk digunakan dalam aplikasi pengesanan *deepfake* yang sebenar.

Penjanaan *Embedding* Wajah Menggunakan Model *FaceNet* Pralatih

Dalam sistem yang dibangunkan, pengecaman wajah hanya diaktifkan sekiranya video input dikenal pasti sebagai video tulen oleh model pengesanan *deepfake*. Ini direka sebagai satu mekanisme keselamatan dua lapis bagi memastikan hanya pengguna sah yang dibenarkan menjalani proses log masuk berdasarkan pengecaman wajah. Untuk tujuan ini, model pengecaman wajah *FaceNet* digunakan. *FaceNet* merupakan model pra-latih yang dilatih ke atas set data *VGGFace2* dan mempunyai keupayaan menghasilkan *embedding* wajah berdimensi 512 yang stabil merentas pelbagai keadaan wajah seperti pencahayaan, ekspresi dan sudut pandang.

Bagi mensimulasikan senario sebenar sistem log masuk pengguna, set data VGGFace2 telah disampel semula untuk membentuk pangkalan data pengguna. Sebanyak 50 individu telah dipilih dan setiap seorang mempunyai kira-kira 100 imej wajah. Set data ini disusun dalam struktur folder standard yang menyokong penggunaan *torchvision.datasets.ImageFolder*, di mana setiap *subfolder* mewakili satu identiti. Proses prapemprosesan imej termasuk penskalaan saiz kepada 160x160 pixel, penukaran ke *tensor*, serta penormalan kepada julat nilai [-1, 1], mengikut keperluan input model *FaceNet*.

Model *FaceNet* dipanggil menggunakan InceptionResnetV1(pretrained='vggface2') dan dijalankan dalam mod inferens (eval) tanpa memerlukan penalaan lanjut. Melalui *DataLoader*, semua imej dihantar secara berkumpulan (*batch*) ke model dan *embedding* wajah diekstrak. *Embedding* ini kemudian dikumpulkan mengikut label kelas (individu), dan bagi setiap pengguna, purata *embedding* dikira. Vektor purata ini digunakan sebagai wakil identiti pengguna, yang memudahkan proses pengecaman secara efisien dan konsisten.

Setelah *embedding* purata bagi semua pengguna diperoleh, ia disimpan dalam satu fail .npy menggunakan *numpy.save*. Fail ini mengandungi satu struktur kamus di mana kunci ialah nama pengguna dan nilai ialah vektor *embedding* 512-dimensi. Fail ini kemudiannya dimasukkan ke dalam sistem *backend*, dan digunakan semasa proses log masuk sebenar. Dalam situasi masa nyata, wajah daripada video input akan diproses oleh *FaceNet*, dan *embedding* yang dijana akan dibandingkan dengan *embedding* daripada fail .json untuk mengenal pasti identiti pengguna berdasarkan jarak atau keserupaan. Proses ini memastikan log masuk hanya dibenarkan kepada pengguna sebenar dengan wajah yang sah, dan pada masa yang sama, menolak sebarang cubaan log masuk yang datang daripada video *deepfake*.

Secara keseluruhannya, pendekatan ini bukan sahaja menyaring video *deepfake* sebelum pengecaman identiti dilakukan, tetapi juga menyediakan satu sistem pengecaman wajah yang cekap, berasaskan *embedding*, dan sesuai untuk digunakan dalam persekitaran masa nyata.

Fasa 5 Penilaian

Fasa penilaian merupakan peringkat kritikal dalam pembangunan model pengesanan *deepfake* kerana ia menentukan keberkesanan model yang dibangunkan dalam mengenal pasti *deepfake* dengan ketepatan tinggi serta kecekapan pemprosesan yang optimum.

Model akan dinilai berdasarkan beberapa metrik utama, iaitu ketepatan (*accuracy*), skor F1 (*F1-score*), kepekaan (*recall*) dan ketepatan klasifikasi (*precision*). Ketepatan akan menunjukkan kadar keseluruhan ramalan yang betul manakala skor F1 akan memberikan keseimbangan antara *precision* dan *recall* bagi menangani kemungkinan ketidakseimbangan dalam data. *Recall* akan menilai sejauh mana model dapat mengesan deepfake dengan betul tanpa terlepas kes positif dan *precision* mengukur ketepatan model dalam mengenal pasti video yang benar-benar *deepfake*. Sekiranya keputusan yang diperoleh tidak mencapai tahap yang memuaskan, proses pembangunan akan dikitar semula ke fasa sebelumnya untuk melakukan penambahbaikan hiperparameter atau pengoptimuman struktur model. Pendekatan ini selaras dengan kaedah pembangunan Agile yang membolehkan proses pembangunan model dilakukan secara berulang tanpa perlu mengulangi semua langkah dari awal kecuali jika terdapat keperluan signifikan untuk perubahan asas.

Fasa 6 Pembangunan

Fasa terakhir iaitu fasa pembangunan merangkumi perancangan dan pelaksanaan pembangunan model yang telah dipilih daripada fasa sebelumnya ke dalam laman web CSAM. Pecutan terakhir ini menitikberatkan pembangunan antara muka untuk integrasi sistem pengesahan *deepfake*, termasuklah pengesahan pengguna dan penerimaan input video daripada pengguna. Selepas model pengecaman *deepfake* diintegrasikan ke dalam platform laman web, ujian akhir akan dijalankan untuk memastikan kelancaran fungsi dan prestasi sistem untuk penggunaan umum. Bahasa HTML, CSS dan JavaScript digunakan bersama API Flask bagi membangunkan laman web ini. Hasil daripada pecutan ini terdiri daripada platform web CSAM yang telah dibangunkan dengan sistem pengesahan *deepfake* yang telah diintegrasikan.

Pembangunan antara muka front-end

Antara muka pengguna dibina menggunakan HTML5, CSS3 dan *JavaScript* bagi menyediakan pengalaman pengguna yang responsif dan interaktif. Struktur halaman menggunakan elemen-elemen semantik HTML5 dengan susun atur *Flexbox* dan *media queries* bagi memastikan paparan sesuai pada pelbagai peranti. Bahagian visual diperkemaskan menggunakan CSS3 dengan penggunaan *gradient background*, animasi *keyframes* dan *backdrop filter blur* untuk memberikan sentuhan profesional dan moden pada antaramuka.

```
async function recordVideo(duration) {
    return new Promise((resolve) => {
        const mediaRecorder = new MediaRecorder(stream, {
            mimeType: 'video/webm;codecs=vp9'
        });
```

Rajah 1.5.1 Fungsi recordVideo()

Bahagian *JavaScript* dalam halaman ini memainkan peranan penting dalam menguruskan fungsi-fungsi interaktif. Sistem menggunakan WebRTC API seperti yang ditunjukkan pada Rajah 1.5.1 untuk mengakses kamera pengguna secara langsung melalui pelayar bagi membolehkan rakaman video wajah pengguna tanpa memerlukan aplikasi luaran. Rakaman video pendek dilakukan menggunakan *MediaRecorder API* membolehkan sistem merakam video wajah pengguna secara automatik untuk dianalisis dalam proses pengesanan *deepfake*. Video yang dirakam akan dihantar ke pelayan *backend* menggunakan *Fetch API* secara *asynchronous* untuk diproses oleh model pengesanan *deepfake* sebelum proses pengesahan wajah diteruskan. Selain itu, *Session Storage API* digunakan untuk menyimpan status pengesahan *deepfake* dan cap masa bagi memastikan pengguna tidak perlu mengulangi proses pengesanan sekiranya mereka telah melepasi pengesanan dalam tempoh masa yang ditetapkan.

Pembangunan back-end

Bahagian *backend* sistem ini dibangunkan menggunakan bahasa pengaturcaraan *Python* dengan *Flask* sebagai kerangka kerja web utama. *Flask* dipilih kerana ringan, fleksibel dan mudah disepadukan dengan model pembelajaran mendalam, membolehkan

pembangunan API *server-side* yang responsif bagi pengendalian proses pengesahan wajah dan pengesanan deepfake secara automatik.

Sistem *backend* ini memanfaatkan *OpenCV* bagi tugasan visi komputer seperti pengesanan wajah dan pemprosesan imej, serta menggunakan *PyTorch* dan *torchvision* untuk operasi pembelajaran mendalam. Model pengecaman wajah menggunakan *FaceNet (facenet-pytorch)* digunakan untuk penjanaan *embedding* wajah pengguna dan perbandingan *embedding* untuk tujuan pengecaman semasa proses pengesahan identiti. Selain itu, *NumPy* dan PIL digunakan untuk manipulasi dan transformasi imej serta operasi matriks dengan cekap semasa prapemprosesan data.

Sistem menguruskan pelbagai *endpoint* API termasuk pendaftaran pengguna, pengesahan wajah, pengesanan *deepfake*, pengurusan *embedding* dan pemeriksaan status pelayan. Kod ini menunjukkan pemanggilan model pengecaman *deepfake* semasa permulaan server Flask, memastikan model hanya dimuat sekali untuk mengoptimumkan penggunaan memori dan masa inferens semasa menerima permintaan daripada pengguna.

```
UPLOAD_FOLDER = 'uploads'

app = Flask(__name__)
CORS(app)  # Enable CORS for all routes
app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER

# Load model once when Flask starts
model = Model()
state_dict = torch.load("deepfake_model.pth", map_location='cpu')
model.load_state_dict(state_dict, strict=False)
model.eval()
```

Rajah 1.5.2 Pecahan kod Modul Deepfake menggunakan Flask API

```python
@app.route("/api/deepfake-detection", methods=["POST"])
def deepfake_detection():
    if 'file' not in request.files:
        return jsonify({"error": "No file provided"}), 400

    file = request.files['file']
    if file.filename == '':
        return jsonify({"error": "No file selected"}), 400

    try:
        if file.filename is None:
            return jsonify({"error": "Invalid filename"}), 400
        filepath = os.path.join(app.config['UPLOAD_FOLDER'], file.filename)
        file.save(filepath)

        # Analyze for deepfake
        label, confidence, probs = analyze_video(filepath, model)
        is_fake = label == "FAKE"

        if is_fake:
            return jsonify({
                "is_real": False,
                "confidence": float(confidence * 100),
                "reason": "Deepfake detected in video"
            })
        else:
            return jsonify({
                "is_real": True,
                "confidence": float(confidence * 100)
            })

    except Exception as e:
        return jsonify({"error": str(e)}), 500
```

Rajah 1.5.3 Fungsi deepfake_detection()

Endpoint ini menerima video daripada pengguna, menjalankan analisis *deepfake* menggunakan model *PyTorch*, dan memulangkan keputusan pengesanan kepada antaramuka pengguna secara *realtime*.

```python
@app.route("/api/register", methods=["POST"])
def register():
    if 'file' not in request.files:
        return jsonify({"error": "No file provided"}), 400

    username = request.form.get('username')
    if not username:
        return jsonify({"error": "Username required"}), 400

    file = request.files['file']
    if file.filename == '':
        return jsonify({"error": "No file selected"}), 400
```

Rajah 1.5.4 Fungsi register()

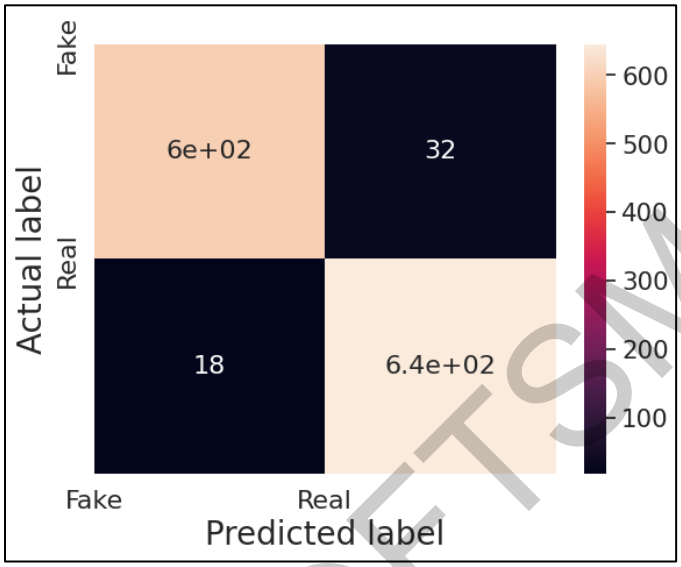
Endpoint ini membolehkan pengguna baharu mendaftar dengan menghantar video wajah, mengekstrak *embedding* menggunakan *FaceNet* dan menyimpan *embedding* dalam format JSON bagi kegunaan semasa proses pengecaman.

```python
@app.route("/api/face-authenticate", methods=["POST"])
def face_authenticate():
    if 'file' not in request.files:
        return jsonify({"error": "No file provided"}), 400

    file = request.files['file']
    if file.filename == '':
        return jsonify({"error": "No file selected"}), 400

    try:
        if file.filename is None:
            return jsonify({"error": "Invalid filename"}), 400
        filepath = os.path.join(app.config['UPLOAD_FOLDER'], file.filename)
        file.save(filepath)

        # Only recognize face (no deepfake check since it's done upfront)
        recognized_user = recognize_face(filepath)
        if recognized_user:
            return jsonify({
                "status": "authenticated",
                "user": recognized_user
            })
        else:
            return jsonify({
                "status": "rejected",
                "reason": "User not recognized"
            })

    except Exception as e:
        return jsonify({"error": str(e)}), 500
```

Rajah 1.5.5 Fungsi face_authenticate()

Endpoint ini akan mengesahkan sama ada video pengguna adalah *deepfake* terlebih dahulu sebelum meneruskan pengecaman wajah. Jika video bukan *deepfake*, sistem akan menjalankan pengecaman wajah bagi mengesahkan identiti pengguna.

KEPUTUSAN PENGUJIAN



Rajah 2.1 Metriks Kekeliruan Model Pengesahan Deepfake

Berdasarkan metriks kekeliruan yang diperoleh daripada pengujian model, nilai *True Positive (TP), True Negative (TN), False Positive (FP) dan False Negative (FN)* telah direkodkan bagi menilai prestasi sistem dalam pengesanan video *deepfake*. Jadual di bawah menunjukkan nilai bagi setiap kategori dalam matriks kekeliruan yang digunakan untuk mengira metrik penilaian seperti ketepatan (*accuracy*), ketepatan positif (*precision*), kepekaan (*recall*) dan skor F1 (*F1-score*).

Jadual 1.1 Jumlah Komponen Metriks Kekeliruan

Kategori	**Penerangan**	**Nilai**
TP (True Positive)	Video deepfake yang berjaya dikenalpasti sebagai *fake* oleh model.	597
TN (True Negative)	Video sebenar yang berjaya dikenalpasti sebagai *real* oleh model.	645
FP (False Positive)	Video sebenar yang salah dikenalpasti sebagai *fake* oleh model.	32
FN (False Negative)	Video deepfake yang salah dikenalpasti sebagai *real* oleh model.	18

Hasil ini menunjukkan bahawa model mempunyai kadar ketepatan pengesanan yang tinggi, memandangkan jumlah kes yang diklasifikasikan dengan betul (TP + TN) adalah sebanyak 1242 daripada keseluruhan 1292 sampel yang diuji. Nilai *False*

Positive yang lebih tinggi berbanding *False Negative* menunjukkan bahawa model cenderung untuk membuat kesilapan dengan mengklasifikasikan video sebenar sebagai *deepfake* berbanding kesilapan dalam mengklasifikasikan *deepfake* sebagai video sebenar. Keadaan ini adalah lebih baik dari perspektif keselamatan sistem pengesahan kerana sistem akan lebih berhati-hati dalam mengenal pasti video sebenar sebelum membenarkan proses pengecaman wajah dilakukan.

Secara keseluruhannya, analisis matriks kekeliruan ini membuktikan bahawa model ResNet-50 + BiLSTM berupaya untuk menjalankan tugas pengesanan *deepfake* dengan baik dan stabil, dengan kadar ralat yang rendah. Keputusan ini mengukuhkan keberkesanan model yang dibangunkan dalam mencapai objektif sistem pengesahan video, di mana hanya video yang tulen dibenarkan melalui proses pengecaman wajah untuk log masuk ke dalam sistem.

Jadual 1.2 Metrik Penilaian Model Pengesahan Deepfake

Metrik Penilaian	**Nilai (%)**
Ketepatan (Accuracy)	96.13
Ketepatan Positif (Precision)	94.91
Kepekaan (Recall)	97.07
Skor F1 (F1-score)	95.98

Berdasarkan hasil ujian model ResNet-50 + BiLSTM menggunakan set data Celeb-DF v2 dan FaceForensics++, nilai ketepatan keseluruhan yang diperoleh adalah sebanyak 96.13%. Nilai ini menunjukkan bahawa model dapat mengklasifikasikan video *deepfake* dan video sebenar dengan tepat dalam sebahagian besar sampel yang diuji. Nilai ketepatan yang tinggi ini menunjukkan bahawa model berupaya mempelajari ciri-ciri penting untuk membezakan antara video *deepfake* dan video sebenar dengan baik, seterusnya dapat membantu dalam proses pengesahan video sebelum pengecaman wajah dilakukan.

Ketepatan positif (*precision*) yang diperoleh adalah 94.91%, yang menunjukkan bahawa daripada semua video yang telah dikesan sebagai *deepfake* oleh sistem, hampir kesemuanya adalah benar-benar *deepfake*. Nilai *precision* yang tinggi ini penting dalam

konteks pengesanan *deepfake* kerana ia menunjukkan sistem dapat mengurangkan kadar pengesanan palsu positif *(false positive)* iaitu kes di mana video sebenar salah dikesan sebagai *deepfake*. Ini menjadikan sistem lebih dipercayai oleh pengguna kerana tidak memberikan amaran palsu yang boleh mengganggu proses pengesahan pengguna.

Kepekaan (recall) sistem adalah 97.07%, yang menunjukkan bahawa sistem mempunyai keupayaan yang sangat baik dalam mengesan video *deepfake* yang terdapat dalam dataset ujian. Nilai *recall* yang tinggi menandakan sistem dapat mengurangkan kadar kes terlepas *(false negative)*, iaitu kes video *deepfake* yang tersalah diklasifikasikan sebagai *real*. Ini amat penting dalam sistem keselamatan dan pengesahan identiti kerana kegagalan mengesan video *deepfake* boleh membawa kepada risiko keselamatan.

Skor F1 yang dicapai adalah 95.98%, menunjukkan keseimbangan yang baik antara ketepatan positif dan kepekaan sistem. Nilai skor F1 yang tinggi ini menjadi indikator penting bahawa sistem bukan sahaja dapat mengesan hampir semua video *deepfake*, tetapi juga dapat memastikan bahawa video yang dikesan sebagai *deepfake* adalah tepat, menjadikan sistem stabil dan konsisten dalam pengesanan.

Secara keseluruhannya, gabungan nilai metrik ini membuktikan bahawa sistem yang dibangunkan bukan sahaja mencapai ketepatan tinggi, tetapi juga mempunyai keseimbangan yang baik dalam mengendalikan pengesanan *deepfake*, mengurangkan kadar kesilapan positif palsu dan negatif palsu, serta meningkatkan kebolehpercayaan sistem apabila digunakan dalam persekitaran sebenar untuk tujuan pengesahan pengguna di laman web CSAM.

Keputusan Pengujian Kotak Hitam

Jadual 1.3 menunjukkan keputusan bagi setiap kes pengujian individu yang telah dijalankan terhadap fungsi utama dalam sistem CSAM. Setiap ujian dilaksanakan berdasarkan spesifikasi yang telah ditetapkan dan menggunakan pendekatan ujian kotak hitam. Hasil pengujian menunjukkan bahawa kesemua fungsi yang diuji telah memberikan output yang menepati jangkaan. Ini termasuk fungsi pendaftaran pengguna baharu, log masuk dengan pengesahan identiti melalui pengesahan *deepfake* dan

pengecaman muka, serta fungsi berkaitan profil seperti paparan, pengemaskinian dan log keluar. Tiada ralat kritikal dikesan semasa ujian dijalankan dan semua kes pengujian telah mencapai status Lulus.

Jadual 1.3 Keputusan pengujian kotak hitam

ID Kes Ujian	ID Fungsi	ID Prosedur Pengujian	Fungsi yang Diuji	Keputusan
P-001	UC-1	PU-001	Pendaftaran akaun baharu	Lulus
P-002	UC-2	PU-002	Log masuk dengan pengesahan identiti	Lulus
P-003	UC-3	PU-003	Pengesahan deepfake	Lulus
P-004	UC-4	PU-004	Pengecaman muka	Lulus
P-005	UC-5	PU-005	Paparan maklumat profil pengguna	Lulus
P-006	UC-6	PU-006	Pengemaskinian profil pengguna	Lulus
P-007	UC-7	PU-007	Log keluar dari sistem	Lulus

Keputusan ini membuktikan bahawa sistem berfungsi dengan baik dari sudut kefungsian dan bersedia untuk digunakan oleh pengguna akhir dalam persekitaran sebenar.

KESIMPULAN

Secara keseluruhannya, projek ini telah berjaya membangunkan sebuah sistem pengesahan identiti berasaskan pengecaman *deepfake* dan pengecaman wajah yang menggabungkan model ResNet-50 dan *Bidirectional Long Short-Term Memory* (BiLSTM). Gabungan model ini membolehkan analisis ciri-ciri spatial dan temporal dalam video dilakukan secara bersepadu bagi mengenal pasti manipulasi wajah yang dihasilkan oleh teknologi *deepfake*.

Sistem ini telah menunjukkan prestasi yang memuaskan dalam persekitaran simulasi, dengan keupayaan untuk mengesan kandungan *deepfake* serta mengesahkan identiti pengguna berdasarkan *embedding* wajah yang dijana. Melalui pendekatan hibrid ini, sistem bukan sahaja dapat meningkatkan keselamatan autentikasi identiti, malah

mampu mengatasi kelemahan sistem pengesahan tradisional yang hanya bergantung kepada kata laluan atau pengecaman wajah secara tunggal.

Walau bagaimanapun, beberapa kekangan telah dikenalpasti termasuk pemprosesan yang tidak masa nyata, risiko keselamatan pada *embedding* biometrik dan penalaan hiperparameter yang terhad akibat kekangan sumber pemprosesan. Kekangan ini memberi gambaran jelas tentang keperluan penambahbaikan sistem untuk diaplikasikan secara lebih meluas dalam persekitaran sebenar.

Cadangan masa hadapan telah dikemukakan bagi mempertingkatkan sistem dari aspek prestasi, kebolehgunaan dan keselamatan. Ini termasuk pelaksanaan pemprosesan masa nyata, perlindungan data biometrik, serta penilaian dalam senario dunia sebenar. Dengan penambahbaikan yang berterusan, sistem ini berpotensi menjadi satu mekanisme pengesahan identiti yang lebih selamat dan berdaya saing dalam menangani cabaran teknologi *deepfake* yang semakin berkembang pesat.

PTA-FTSM-2025-A195635

RUJUKAN

A Review of Deep Learning-based Approaches for Deepfake Content Detection. 2024. . *Elsevier*.

Al-Dulaimi, O.A.H.H. & Kurnaz, S. 2024. A hybrid CNN-LSTM approach for precision deepfake image detection based on transfer learning. Electronics 13(9): 1662.

Arun, M., Deenadhayalan, M., Deepak, R.U., Balaraman, S. & Deepak, D. 2024. Lung Cancer Detection using ResNet-50 CNN Architecture: 1–6.

Arya, M., Priyanshu, N., Upwan, N., Akash, N., Goyal, U. & Chawla, S. 2024. A Study on Deep Fake Face Detection Techniques. 2024 3rd International Conference on Applied Artificial Intelligence and Computing (ICAAIC): 459–466.

Berroukham, A., Housni, K. & Lahraichi, M. 2023. Vision Transformers: A review of architecture, applications, and future directions. 2023 7th IEEE Congress on Information Science and Technology (CiSt): 205–210.

Chen, Z., Wang, S., Yan, D. & Li, Y. 2024. A Spatio- Temporl Deepfake Video Detection Method Based on TimeSformer-CNN.

Dagar, D. & Vishwakarma, D.K. 2023b. A Hybrid Xception-LSTM Model with Channel and Spatial Attention Mechanism for Deepfake Video Detection. *2023 3rd International Conference on Mobile Networks and Wireless Communications*: 1–5.

Garg, D. & Gill, R. 2023. Deepfake Generation and Detection - an exploratory study. 2018 5th IEEE Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON): 888–893.

Gong, D., Kumar, Y.J., Goh, O.S., Ye, Z. & Chi, W. 2021b. DeepfakeNet, an efficient deepfake detection method. *International Journal of Advanced Computer Science and Applications* 12(6).

Jakka, A., Rani, V., Challa, M., Kumar, M.V. & Kookkal, G. 2024b. Deepfake Video Detection using Deep Learning Approach. *2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT)*: 1–6.

Janiesch, C., Zschech, P. & Heinrich, K. 2021b. Machine learning and deep learning. *Electronic Markets* 31(3): 685–695.

John, J. & Sherif, B.V. 2022. Comparative analysis on different DeepFake detection methods and semi supervised GAN architecture for DeepFake detection. 2022 Sixth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC) 3: 516–521.

Johri, P. & Arora, S. 2022. Review of the issues and a thorough investigation of biometric authentication systems. 2022 International Interdisciplinary Humanitarian Conference for Sustainability (IIHC): 892–897.

PTA-FTSM-2025-A195635

Keerthana, S., Deepika, N., Pooja, Er., Nandhini, I., Shanthalakshmi, M. & Khanaghavalle, G.R. 2024. An effective approach for detecting deepfake videos using Long Short-Term Memory and ResNet.

Lee, K., Jung, I. & Woo, S.S. 2024. iFakeDetector: Real Time Integrated Web-based Deepfake Detection System. *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*: 8717–8720.

Patel, S., Chandra, S.K. & Jain, A. 2023. DeepFake Videos Detection and Classification Using Resnext and LSTM Neural Network.

Pipin, S., Purba, R. & Pasha, M.F. (pnyt.). 2022. Deepfake Video Detection Using Spatiotemporal Convolutional Network and Photo Response Non Uniformity. 2022 IEEE International Conference of Computer Science and Information Technology (ICOSNIKOM).

Pipin, S., Purba, R. & Pasha, M.F. 2022. Deepfake Video Detection Using Spatiotemporal Convolutional Network and Photo Response Non Uniformity. 2022 IEEE International Conference of Computer Science and Information Technology (ICOSNIKOM).

Pipin, S.J., Purba, R. & Pasha, M.F. 2022b. Deepfake Video Detection Using Spatiotemporal Convolutional Network and Photo Response Non Uniformity. *2022 IEEE International Conference of Computer Science and Information Technology*: 1–6.

Rebello, L., Tuscano, L., Shah, Y., Solomon, A. & Shrivastava, V. 2023d. Detection of Deepfake Video using Deep Learning and MesoNet. *2022 7th International Conference on Communication and Electronics Systems (ICCES)*: 1022–1026.

Salsabila, Z.H., Nurmalasari, R.R. & Kamelia, L. 2024. Indonesian Sign Language Translation System Using ResNet-50 Architecture-Based Convolutional Neural Network: 1–5.

Sharma, V.K., Garg, R. & Caudron, Q. 2023. Spatio-Temporal Convolutional Neural Networks for Deepfake Detection: An Empirical Study. 2023 Second International Conference on Informatics (ICI): 1–7.

Sonkusare, M.G., Meshram, H.A., Sah, A. & Prakash, S. 2022. Detection and verification for deepfake bypassed facial feature authentication. 2022 Second International Conference on Artificial Intelligence and Smart Energy (ICAIS): 646–649.

Subramanian, M., S, L.S. & R, R.V. 2023. Deep Learning Approaches for Melody Generation: An evaluation using LSTM, BILSTM and GRU models. 2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT).

Sun, R., Zhao, Z., Shen, L., Zeng, Z., Li, Y., Veeravalli, B. & Xulei, Y. 2023b. An efficient deep video model for deepfake detection. *2022 IEEE International Conference on Image Processing (ICIP)*: 351–355.

Sundaram, V., Senthil, B. & Vekkot, S. 2024c. Enhancing deepfake detection: leveraging deep models for video authentication. *2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT)*: 1–7.

PTA-FTSM-2025-A195635

Taviti, R., Taviti, S., Reddy, P.A.K., Sankar, N.R., Veneela, T. & Goud, P.B. 2023. Detecting Deepfakes With ResNext and LSTM: An Enhanced Feature Extraction and Classification Framework.

Waseem, S., Bakar, S.A.R.S.A., Ahmed, B.A., Omar, Z., Eisa, T.A.E. & Dalam, M.E.E. 2023. DeepFake on Face and Expression Swap: a review. IEEE Access 11: 117865–117906.

Wazid, M., Mishra, A.K., Mohd, N. & Das, A.K. 2024b. A secure deepfake mitigation framework: architecture, issues, challenges, and societal impact. *Cyber Security and Applications* 2: 100040.

Zhang, R., Jiang, Z. & Sun, C. 2023b. Two-Branch Deepfake Detection Network Based on Improved Xception. *2023 IEEE International Conference on Electrical, Automation and Computer Engineering*: 227–231.