**Deepfake AI Detection Model**

A project Submitted in partial fulfillment of the requirements for the award of the degree of

**Master of Computer Applications**

**In**

**Computer Applications**

**By**

**Ashay Jain (205121028)**

****

**DEPARTMENT OF COMPUTER APPLICATIONS**

**NATIONAL INSTITUTE OF TECHNOLOGY, TIRUCHIRAPPALLI 620015**

**BONAFIDE CERTIFICATE**

This is to certify that the project **“DeepFake AI Detection Model”** is a project work successfully done by

Ashay Jain (205121028)

in partial fulfillment of the requirements for the award of the degree of Master of Computer Applications from the National Institute of Technology, Tiruchirappalli, during the academic year 2021-2024 (5th Semester – CA749 Mini Project Work).

Dr. SelvaKumar K. Dr. Michael Arock

Project Guide Head of the Department

Project viva-voce held on …………………………….

Acknowledgment

Every project, big or small, is successful largely due to the effort of a number of wonderful people who have always given their valuable advice or lent a helping hand. I sincerely appreciate the inspiration, support, and guidance of all those people who have been instrumental in making this project successful.

We express our deep sense of gratitude to **Dr. G. Aghila**, Director, National Institute of Technology, Tiruchirappalli for giving us an opportunity to do this project.

I am grateful to **Dr. Michael Arock**, Professor, and Head of the Department of Computer Applications, National Institute of Technology, Tiruchirappalli for providing the infrastructure and facilities to carry out the project.

I express my gratitude to my Project Guide **Dr.SelvaKumar K.**, Professor, Department of Computer Applications, National Institute of Technology, Tiruchirappalli for his support and for arranging the project in a good schedule, and who assisted me in completing the project. I would like to thank him for duly evaluating my progress and evaluating me.

I express my sincere and heartfelt gratitude to **Dr. S. Balasundaram**, Mini Project Work Coordinator, Department of Computer Applications, National Institute of Technology, Tiruchirappalli. I am sincerely thankful for his constant support, care, guidance, and regular interaction throughout my project.

I express my sincere thanks to all the faculty members, and scholars of NIT Trichy for their critical advice and guidance to develop this project directly or indirectly.

Abstract

Deepfake technology, leveraging advanced artificial intelligence (AI) and machine learning algorithms, has become a potent tool for creating highly realistic fake videos that pose a threat to trust, privacy, and security. This abstract explores the critical domain of Deepfake Video Detection AI, which focuses on developing sophisticated algorithms and techniques to identify and mitigate the risks associated with deceptive content. The research delves into the underlying principles of deepfake generation, outlines current detection methodologies, and highlights emerging trends in AI-driven countermeasures. By comprehensively examining the landscape of deepfake video detection, this abstract aims to contribute to the ongoing efforts to safeguard individuals, organizations, and societies from the malicious applications of this evolving technology. The insights provided herein underscore the importance of continuous research, collaborative initiatives, and ethical considerations in the development of robust AI solutions to combat the challenges posed by deepfake videos.

Table of Contents

Title Page no.

Bonafide Certificate…………………………………………………………………..............1

Acknowledgement……………………………………………………………………………2

Abstract……………………………………………………………………………………….3

Table of Contents……………………………………………………………………………..4

Table of Figures………………………………………………………………………………5

Chapter 1 : Introduction ……………………………………………………………………...6

Chapter 2 : How DeepFake Works

2.1: Deep Learning Algorithm

2.2: Data Collection and Training

2.3: Generation of Fake content

Chapter 3 : Applications of DeepFake Technology

3.1: Entertainment Industry

3.2: Political Manipulation

3.3: Fraud and Cyber Security risk

3.4:Positive Use case

Chapter 4 : Dataset

Chapter 5 : Methodologies

5.1 LSTM

5.2 ResNext

5.3 RNN

Chapter 6 : Project Working and Explanation

Chapter 7 : System Design

7.1: Architecture

7.2: Flow Diagram

Chapter 8 : Challenges and Risks

8.1: Misinformation and Fake News

8.2: Privacy Concerns

8.3: Legal and Ethical information

8.4: Technological Challenges

Chapter 9 : Results and Discussion

Chapter 10 : Conclusion

Table Of Figures

Fig no. Title Page no.

4.1 Dataset

4.2 Preprocessing

4.3 Model Architecture

5.1 Methodologies LSTM

7.1 Architecture

7.2 Flow Diagram

9.1 Results

1. Introduction

### 1.1 Definition and Origins:

Deepfake technology, a term originating from the amalgamation of "deep learning" and "fake," emerged during the mid-2010s as a direct consequence of significant advancements in the fields of deep learning and neural networks. At its core, deepfake technology represents the utilization of sophisticated algorithms to manipulate or generate visual and audio content that possesses a strikingly realistic appearance. The term "deep" underscores the reliance on deep learning techniques, which involve complex neural networks capable of learning intricate patterns and features from extensive datasets.

The essence of deepfake technology lies in its ability to employ artificial intelligence and machine learning to recreate human-like attributes, seamlessly blending fabricated elements into existing content. This synthesis of technology allows for the creation of content that, on the surface, appears indistinguishable from authentic visual or auditory experiences. This capability has profound implications for various fields, from entertainment to more serious considerations like misinformation and privacy infringement.

### 1.2 Evolution of Deepfake Technology:

In its nascent stages, deepfake technology found its foothold in benign applications, notably within the entertainment industry. Early use cases involved the creation of realistic animations and computer-generated characters, showcasing the positive potential of the technology in enhancing visual experiences. However, as deep learning algorithms became more sophisticated, the evolution of deepfake technology took a transformative turn.

The technology progressed beyond its initial applications, giving rise to the creation of highly convincing fake videos. These videos, often featuring individuals saying or doing things they never actually did, raised substantial concerns about the potential for misuse. The evolution of deepfake technology brought to the forefront the ethical and societal challenges associated with the manipulation of visual and auditory information.

This progression prompted a shift in the perception of deepfake technology from a tool for creative expression to a potential threat to truth, authenticity, and privacy. As the technology continues to advance, it necessitates a nuanced understanding of its origins and evolution to address the multifaceted challenges it presents. Balancing innovation with ethical considerations becomes imperative in navigating the complex landscape shaped by the evolution of deepfake technology.

1. How DeepFake Works

### 2.1 Deep Learning Algorithms:

The core functionality of deepfake technology hinges on the intricate capabilities of deep learning algorithms, specifically emphasizing the utilization of generative adversarial networks (GANs) and autoencoders. GANs, a pivotal element in deepfake creation, operate through a dynamic interplay between two neural networks—the generator and the discriminator. The generator crafts synthetic content, while the discriminator evaluates its authenticity. This adversarial process iteratively refines the generated content until it attains a level of realism that is challenging to distinguish from genuine material.

Autoencoders, another class of deep learning algorithms, contribute to the encoding and decoding of information. By compressing input data into a latent representation and then reconstructing it, autoencoders facilitate the extraction of intricate patterns essential for creating convincing deepfake content.

### 2.2 Data Collection and Training:

The genesis of a deepfake involves a meticulous process of data collection and subsequent training of the deep learning model. To achieve a high level of realism, a substantial amount of diverse data is required, encompassing facial expressions, gestures, voice nuances, and other distinctive characteristics associated with the target individual. This dataset serves as the foundation upon which the deep learning algorithm learns the intricacies of the subject's appearance and behavior.

During the training phase, the deepfake model refines its parameters by iteratively processing the dataset. The algorithm adjusts its internal weights and biases to accurately replicate the learned features. This training process is pivotal in enabling the model to generalize its understanding of the target individual, ensuring adaptability to various scenarios and inputs.

### 2.3 Generation of Fake Content:

Once adequately trained, the deepfake algorithm enters the content generation phase. Leveraging the acquired knowledge from the training dataset, the algorithm can seamlessly combine and manipulate features to create synthetic videos or audio recordings. This amalgamation of learned facial expressions, gestures, and voice characteristics results in content that, to the human observer, appears remarkably authentic.

The generated deepfake content is not a mere replication but a creative synthesis that introduces novel elements. Through this process, deepfake technology achieves its goal of producing content that blurs the line between reality and fabrication. The sophistication of the algorithms involved in this phase underscores the continual evolution of deepfake technology and its potential to generate content with increasingly convincing fidelity.

1. Applications Of DeepFake Technology

### 3.1 Entertainment Industry:

In the realm of the entertainment industry, deepfake technology has become a double-edged sword. On the positive side, it offers innovative solutions for creating realistic computer-generated characters and scenes. Filmmakers and animators can utilize deepfake algorithms to enhance visual effects, bringing to life fantastical creatures or seamlessly integrating CGI elements into live-action footage. This application has the potential to revolutionize the way movies and television productions approach special effects, providing a cost-effective and efficient alternative to traditional methods.

However, the positive contributions of deepfake technology in entertainment come with a notable caveat. Concerns have arisen about the possible replacement of human actors by synthetic counterparts. This raises ethical questions about the future of the acting profession, as well as considerations related to the authenticity and emotional depth conveyed by human performers compared to their deepfake counterparts.

### 3.2 Political Manipulation:

Deepfake technology's impact extends beyond the realm of entertainment into the sphere of political manipulation. The ability to create convincing fake videos of political figures introduces a grave risk to public opinion, electoral processes, and international relations. Malicious actors can exploit deepfake technology to fabricate speeches, interviews, or events featuring political figures, potentially swaying public sentiment, sowing discord, or even influencing election outcomes.

The rise of deepfake-based political manipulation poses challenges to the trustworthiness of information disseminated through digital channels. It underscores the need for robust mechanisms to verify the authenticity of audiovisual content, especially in the context of political discourse where the stakes are high.

### 3.3 Fraud and Cybersecurity Risks:

The darker side of deepfake technology manifests in the realm of fraud and cybersecurity risks. Criminals can leverage deepfakes to perpetrate fraud, identity theft, and cyber attacks with heightened sophistication. The ability to create convincing impersonations of individuals, including corporate executives or public figures, poses a significant threat to organizations and individuals alike.

Deepfake-driven cyber attacks may involve manipulating audio or video content to deceive employees into divulging sensitive information, leading to financial losses or unauthorized access to secure systems. As a result, the need for robust cybersecurity measures, employee awareness training, and advanced fraud detection technologies becomes paramount in the face of evolving deepfake threats.

### 3.4 Positive Use Cases:

Despite the inherent risks, deepfake technology also presents positive use cases that contribute to various industries. In the realm of entertainment, deepfakes can improve the quality of dubbing in movies by synchronizing lip movements with translated dialogue, providing a more immersive viewing experience for global audiences.

Moreover, deepfake algorithms can be harnessed to create lifelike characters in video games, enhancing the realism and interactivity of virtual worlds. This application opens new avenues for immersive gaming experiences and storytelling.

Additionally, deepfake technology has the potential to enhance virtual communication by improving video conferencing and virtual reality interactions. Realistic avatars generated through deepfake algorithms can make virtual meetings more engaging and dynamic, fostering a sense of presence and connection in remote communication.

Balancing the positive use cases with the risks and ethical considerations remains a critical challenge, requiring ongoing scrutiny, regulation, and responsible deployment of deepfake technology across diverse sectors.

1. Dataset

The project utilizes a mixed dataset consisting of both genuine and manipulated videos. This dataset is collected from various sources, including YouTube, FaceForensics++, and the DeepFake detection challenge dataset. The dataset is evenly split into training and testing sets

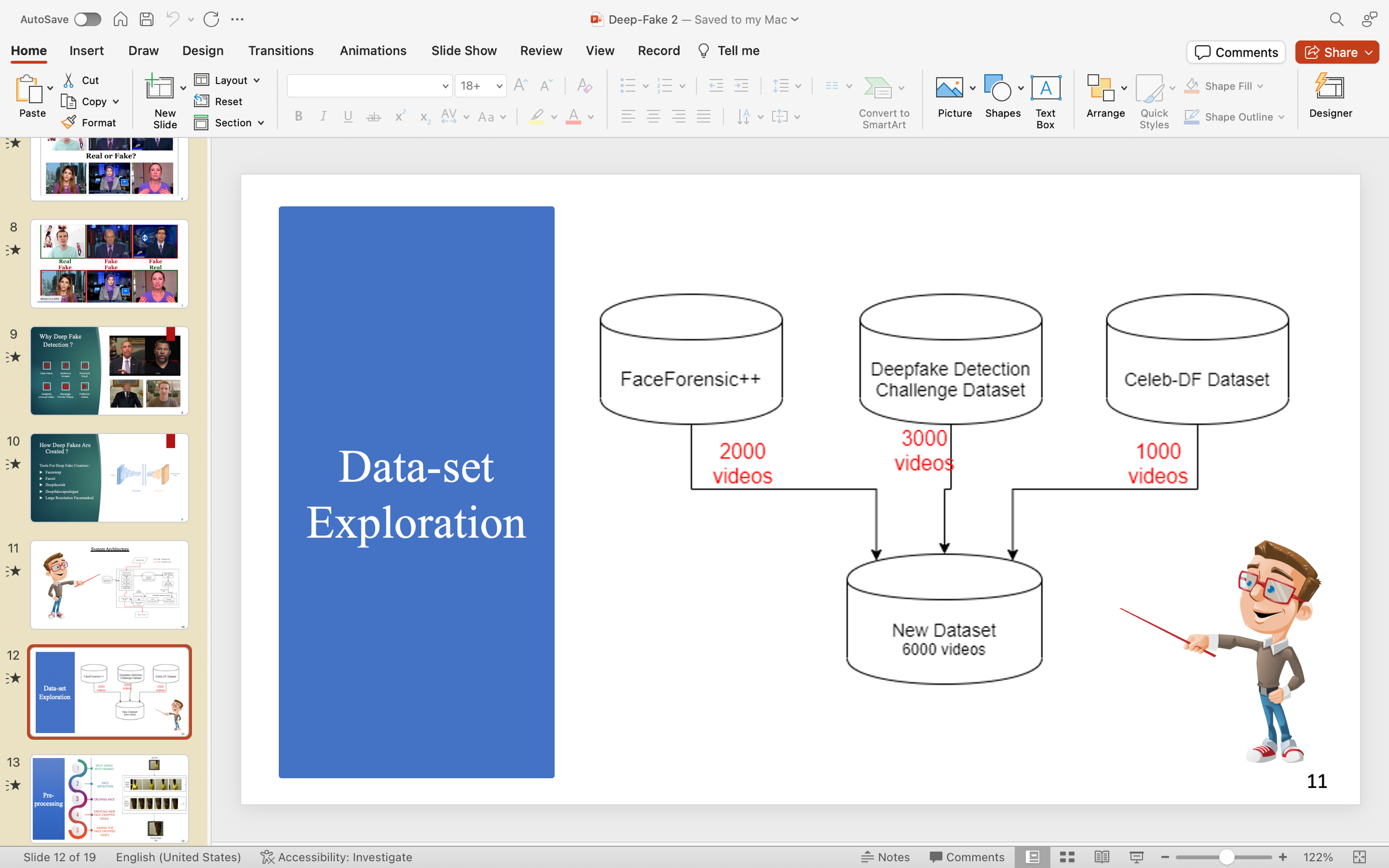


Fig 4.1 Dataset Collection

**Pre-processing:-**

The initial step involves pre-processing the dataset. This includes splitting the videos into frames, detecting faces within the frames, cropping the frames to contain only the detected faces, and ensuring uniformity in the number of frames. Frames without detected faces are discarded. To manage computational resources, only the first 100 frames of a 10-second video (30 frames per second) are used for experimentation.

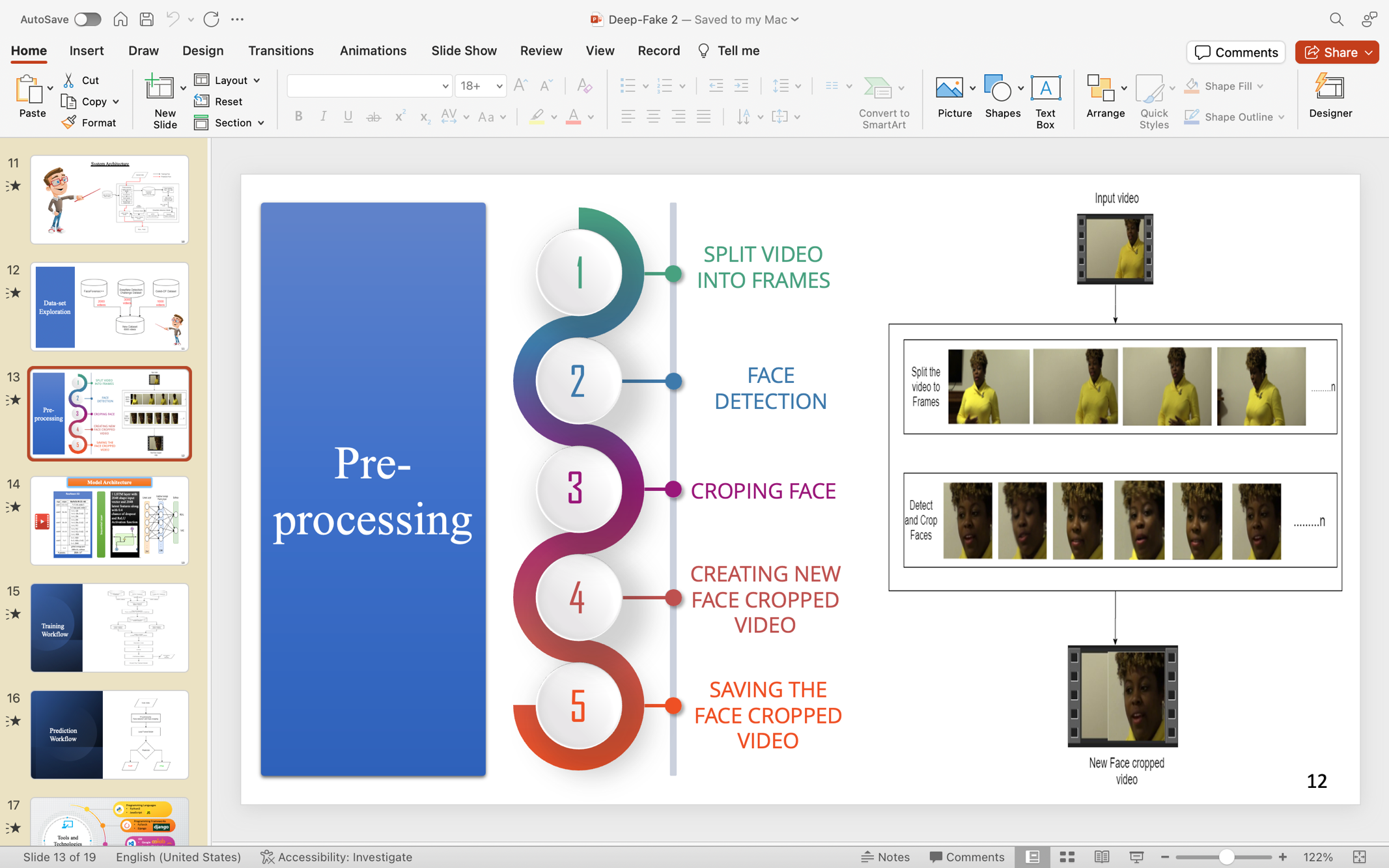


Fig 4.2 Preprocessing

**Model:-**

The model architecture comprises a ResNext50\_32x4d for feature extraction, followed by an LSTM layer for sequence processing. The ResNext CNN is used to extract frame-level features, and the LSTM is employed to analyse temporal inconsistencies between frames introduced by the DeepFake creation process.

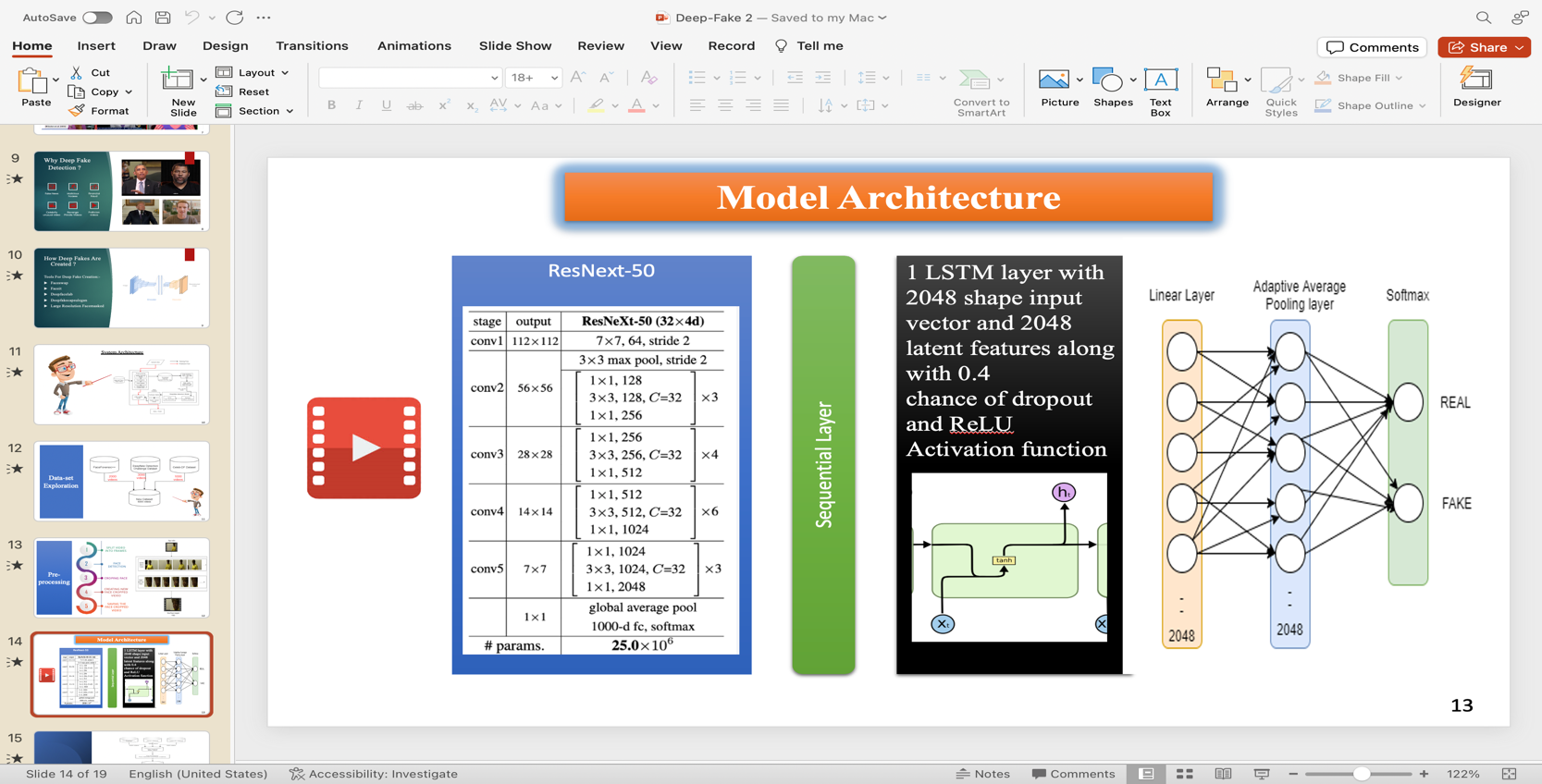


Fig 4.3 Model Architecture

1. Challenges and Risks

### 5.1 Misinformation and Fake News:

The proliferation of deepfake content poses a significant challenge in the form of misinformation and fake news. As deepfake technology enables the creation of highly convincing fabricated videos, there is a heightened risk of malicious actors exploiting this capability to spread false narratives. The potential consequences include the erosion of trust in media, the manipulation of public opinion, and the destabilization of democratic processes. Detecting and countering deepfake-based misinformation requires a concerted effort from both technology developers and media literacy initiatives to empower individuals with the skills to discern between authentic and manipulated content.

### 5.2 Privacy Concerns:

Deepfakes introduce profound privacy concerns, primarily stemming from the unauthorized manipulation of individuals' likeness and personal data. The technology allows for the creation of fabricated videos featuring people engaging in activities they never participated in or saying things they never uttered. This infringes on the individual's right to control their own image and likeness, raising ethical and legal questions about consent. As deepfakes become more sophisticated, the potential for malicious actors to exploit personal data for various purposes, including defamation or extortion, underscores the urgent need for robust privacy protections and legislative measures.

### 5.3 Legal and Ethical Implications:

The rapid evolution of deepfake technology has outpaced the development of comprehensive legal frameworks and ethical guidelines. This lag in regulation creates a complex landscape with uncertainties surrounding issues such as consent, accountability, and responsible use. Ethical considerations become paramount when addressing questions about the boundaries of creative expression, the right to manipulate digital media, and the potential societal impacts of deepfake deployment. Striking a balance between technological innovation and the protection of individual rights requires collaborative efforts from lawmakers, ethicists, and technology developers to establish a framework that safeguards against misuse while fostering responsible innovation.

### 5.4 Technological Challenges:

The cat-and-mouse game between deepfake creators and detection mechanisms presents a perpetual technological challenge. As deepfake algorithms become more sophisticated, traditional detection methods struggle to keep pace. Continuous advancements in deepfake technology, including the integration of adversarial techniques designed to evade detection, create a dynamic landscape for cybersecurity experts. Developing effective detection tools requires ongoing research, collaboration, and innovation to stay ahead of the evolving capabilities of deepfake algorithms. The arms race between those creating deceptive content and those seeking to identify and prevent its dissemination underscores the need for a proactive and adaptive approach to technological challenges in the deepfake landscape.

1. Methodologies
   1. **LSTM :**

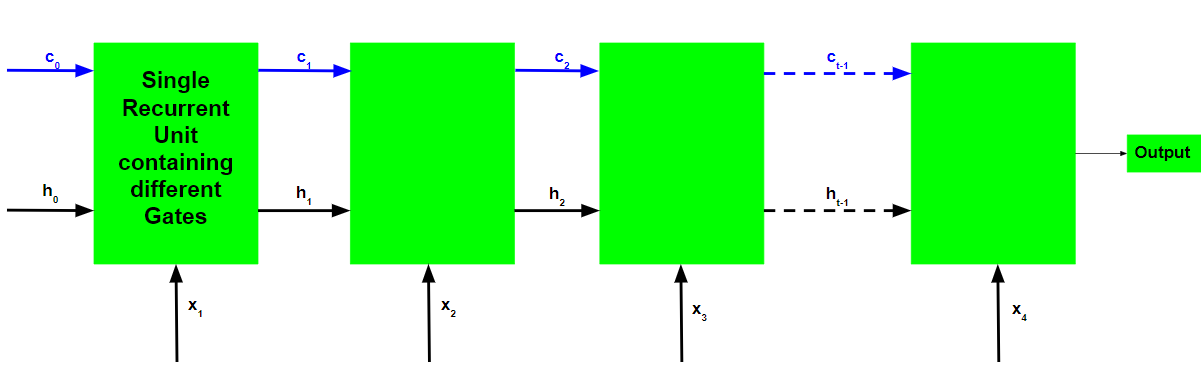


Fig 6.1 LSTM DIAGRAM

1. Project Working and Explanation

The project will follow these steps:

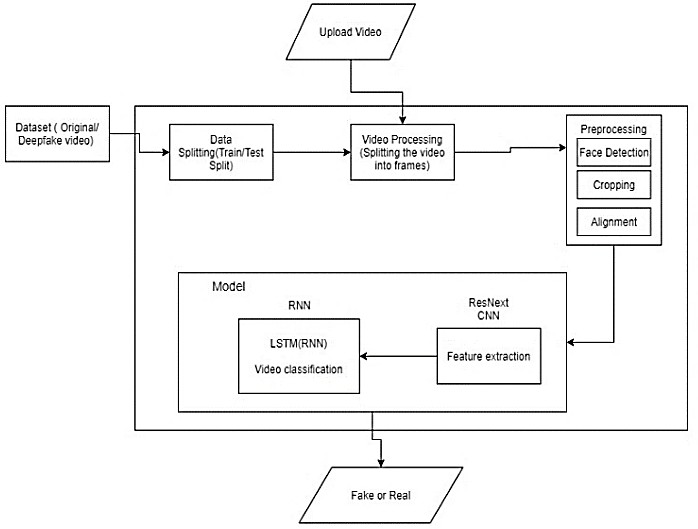
1. Data Collection: Collect a dataset of real and deepfake videos. The dataset should be large enough to train the model effectively.
2. Data Preprocessing: Preprocess the videos to ensure they are in a format that can be used by the CNN model. This may involve resizing the videos, normalizing the pixel values, and extracting relevant features.
3. Model Training: Train the CNN model on the preprocessed data. The model will learn to identify patterns in the videos that are indicative of deepfake manipulation.
4. Model Evaluation: Evaluate the trained model on a separate dataset of real and deepfake videos. The evaluation should measure the model's accuracy in detecting deepfake videos.

**Explanation of Deepfake Video Detection Model**

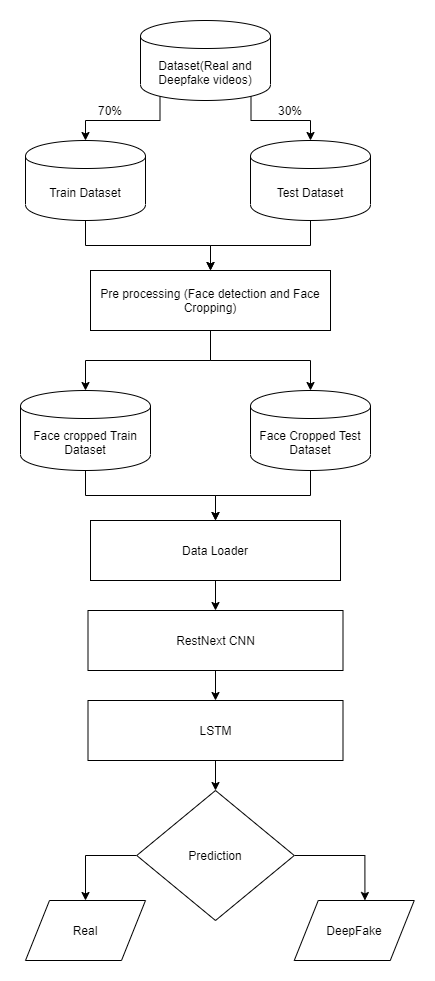
A deepfake video detection model is a machine learning model that is trained to identify deepfake videos. Deepfake videos are videos that have been manipulated using artificial intelligence to make it appear as if someone is saying or doing something that they never actually said or did.

1. System Design

8.1 Architecture



8.2 Flow Diagram



7.RESULTS AND DISCUSSION

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model Name | Dataset | No. of videos | Sequence Length | Accuracy |
|  | FaceForensic++ | 2000 | 20 | 90.95477387 |
|  | 40 | 95.22613065 |
|  | 60 | 97.48743719 |
|  | 80 | 97.73366834 |
|  | 100 | 97.76180905 |
|  | Our Dataset | 6000 | 10 | 84.662519 |
|  | 20 | 87.79160186 |
|  | 40 | 89.34811819 |
|  | 60 | 91.59097978 |
|  | 80 | 92.49818558 |
|  | 100 | 92.10883877 |

8. CONCLUSION

Developing a deepfake video detection model is a critical endeavor in the ongoing battle against the misuse of AI-generated content. As we conclude the exploration of this model, several key takeaways emerge:

1. **Innovation vs. Safeguarding Integrity:**
   * The rapid evolution of deepfake technology necessitates constant innovation in detection methods to safeguard the integrity of digital content.
2. **Multidimensional Challenge:**
   * Deepfake detection is a multidimensional challenge that involves computer vision, machine learning, and cybersecurity. Addressing this challenge requires a holistic approach.
3. **Dataset Diversity Matters:**
   * The effectiveness of a detection model heavily relies on the diversity and representativeness of the dataset used for training. A robust dataset encompassing various deepfake creation techniques is paramount.
4. **Interdisciplinary Collaboration:**
   * Success in developing effective deepfake detection models requires interdisciplinary collaboration between computer scientists, ethicists, psychologists, and legal experts. Ethical considerations and legal frameworks must evolve alongside technological advancements.
5. **Real-time Analysis:**
   * The deployment of deepfake detection models in real-time applications enhances their practical utility. Integrating these models into platforms, social media networks, and other digital spaces can contribute to the prevention of deceptive content dissemination.
6. **Ongoing Research and Adaptability:**
   * The cat-and-mouse game between deepfake creators and detection models necessitates ongoing research and adaptability. Staying ahead of emerging techniques used in deepfake creation requires continuous updates and improvements to detection algorithms.
7. **User Education:**
   * Beyond technological solutions, educating users about the existence and potential risks of deepfake content plays a crucial role. Increased media literacy empowers individuals to critically evaluate the authenticity of digital content.
8. **Balancing Innovation and Regulation:**
   * Striking a balance between fostering innovation in AI technologies and implementing regulatory frameworks to curb malicious use is a delicate task. Collaborative efforts between governments, tech industries, and research institutions are pivotal in achieving this balance.

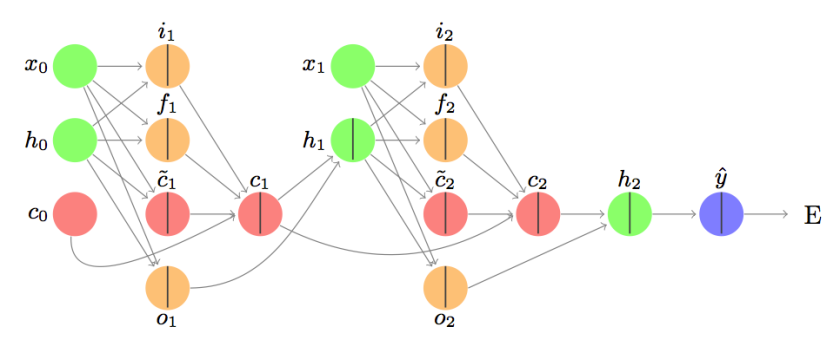
In conclusion, while deepfake technology poses significant challenges, the development and deployment of robust detection models represent a crucial step in mitigating its adverse effects. The journey toward effective deepfake detection involves continuous learning, adaptation, and a commitment to ethical practices in the rapidly evolving landscape of AI-generated content. The collective efforts of researchers, industry stakeholders, policymakers, and the public are essential to navigate this complex and dynamic terrain successfully.

1. Comparision

**Our Model VS Existing Models**

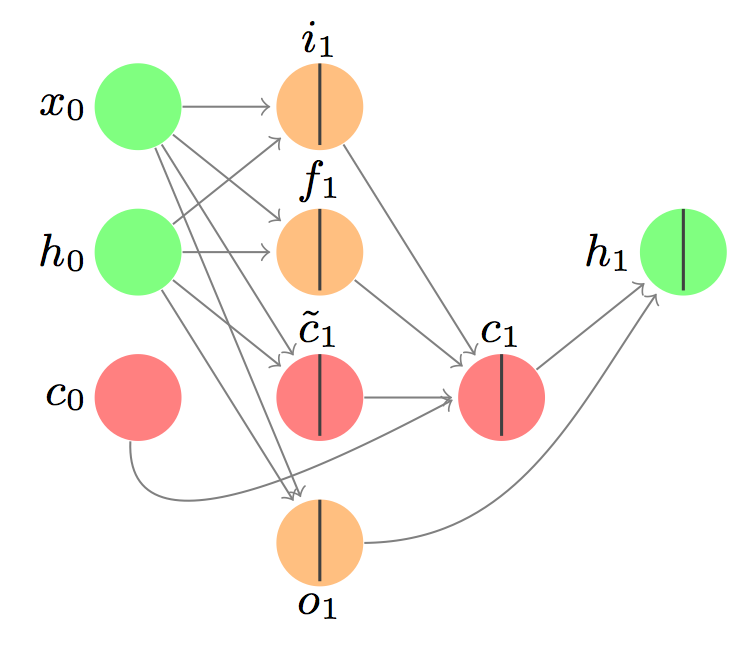
|  |  |
| --- | --- |
| Our proposed system distinguishes itself by combining CNN and RNN to capture temporal inconsistencies, providing a robust and competitive solution | Existing approaches for deepfake detection use capsule networks. |
| Unlike some existing models, our method considers various parameters such as teeth enhancement and wrinkles for a comprehensive detection approach. | Existing models use methods such as detecting face warping artifacts, eye blinking. |
| Our proposed system stands out as it not only detects deepfake videos but also offers a scalable solution, from a web-based platform to integration with popular applications like WhatsApp and Facebook. | While various tools exist for creating deepfake content, the detection tools are limited |
| Pixel by pixel division of frames with different sizes like 2x2, 4x4, 8x8 is done. | Fixed size frames are superimposed. |
| Original Image is not needed to predict the result. | Original Image must be given in the beginning for the reference. |

NUMERICAL EXPALINATION:-

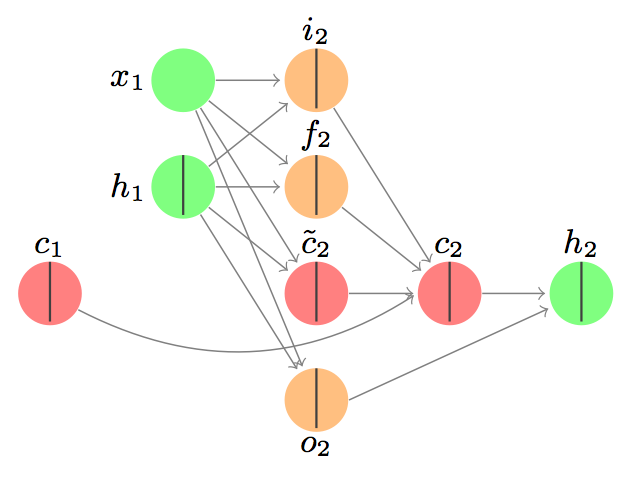
**This is an LSTM with two cells:**  


Let’s break that image down a bit.

Here is the first cell:



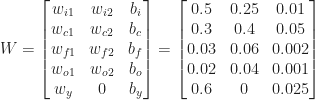
Here is the second cell:

**A detailed walk-through: forward**

Let’s start with a simple example with one dimensional input and one dimensional output.

Let’s focus on the first cell for now:

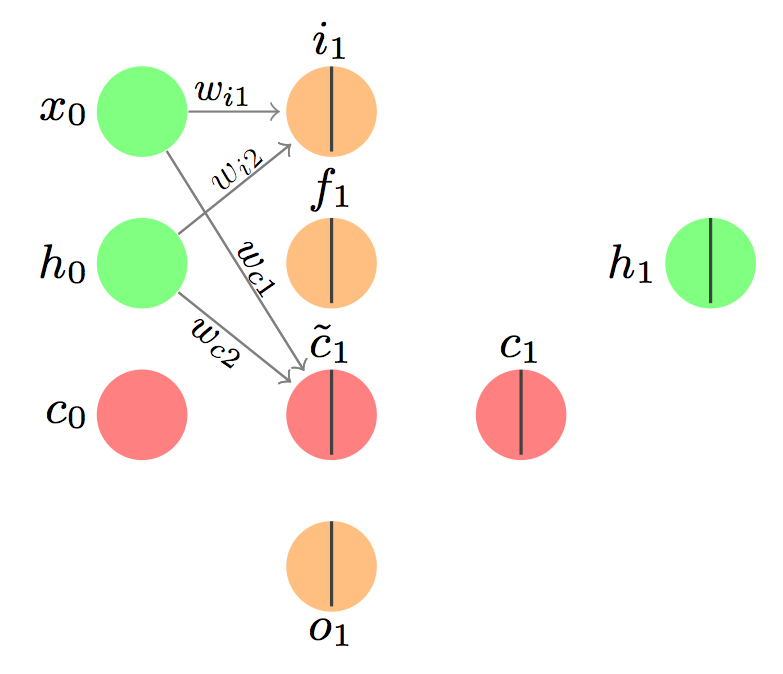
Suppose we have a scalar-valued input sequence x_0 = 0.1, x_1 = 0.2. In English, this means the input at the beginning of the sequence is 0.1, and the input at the next time step is 0.2.



This formulation will come in handy later for backpropagation, but you can see that each row of the matrix W has all of the parameters needed for one of the gates. The last row is the linear transformation associated with the output (we’ll get to that).

### The input gate

The image associated with the input gate is:



And the associated equations for the first part are:

{\rm{net}}_{i1} = w_{i1}x_1 + w_{i2}h_0 + b_i

i_1 = \sigma({\rm{net}}_{i1}) = 1/(1 + exp(-{\rm{net}}_{i1} ))

I’ve used ‘net’ to mean the net input to the gate. We take a linear transformation of the input values. Another way to present the linear transformation (using T for transpose) is:

{\rm{net}}_{i1} = [w_{i1} \hspace{1em} w_{i2}] [x_1 \hspace{1em} h_0]^T + b_i, as done on that first blog I linked to.

The full computation is:

{\rm{net}}_{i1} = 0.5(0.1) + 0.25(0) + 0.01 = 0.06

i_1 = \sigma({\rm{net}}_{i1}) = 1/(1 + exp(-0.06)) = 0.515

This value can be interpreted as the probability that we will allow the information from x_1 to enter the memory cell.

{\rm{net}}_{c1} = w_{c1}x_1 + w_{c2}h_0 + b_c

\tilde{c}_1 = \sigma({\rm{net}}_{c1}) = 1/(1 + exp(-{\rm{net}}_{c1} ))

The full computation is:

{\rm{net}}_{c1} = 0.3(0.1) + 0.4(0) + 0.05 = 0.08

\tilde{c}_1 = \tanh({\rm{net}}_{c1}) = 1/(1 + exp(-0.08)) = 0.0798

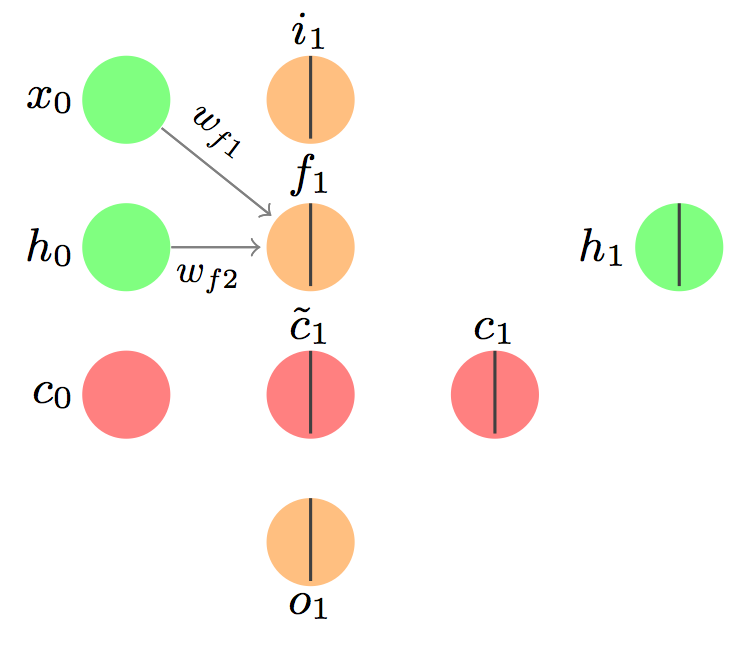
Note no stochastic decision is made here – this is the quantity associated with the input that we'll pass to the memory cell. We could make a stochastic decision using a tanh function, and that often happens, but not here. Why? Because this is the input signal! We need this part as it is.

We’ll use both of these pieces together later when we update the memory cell.

### The forget gate

The point of this gate is to decide what information needs to be removed from the network.

The forget gate looks like this:



and takes similar input:

{\rm{net}}_{f1} = w_{f1}x_1 + w_{f2}h_0 + b_f

f_1 = \sigma({\rm{net}}_{f1}) = 1/(1 + exp(-{\rm{net}}_{f1} ))

And the computation is also similar:

{\rm{net}}_{f1} = 0.03(0.1) + 0.06(0) + 0.002 = 0.005

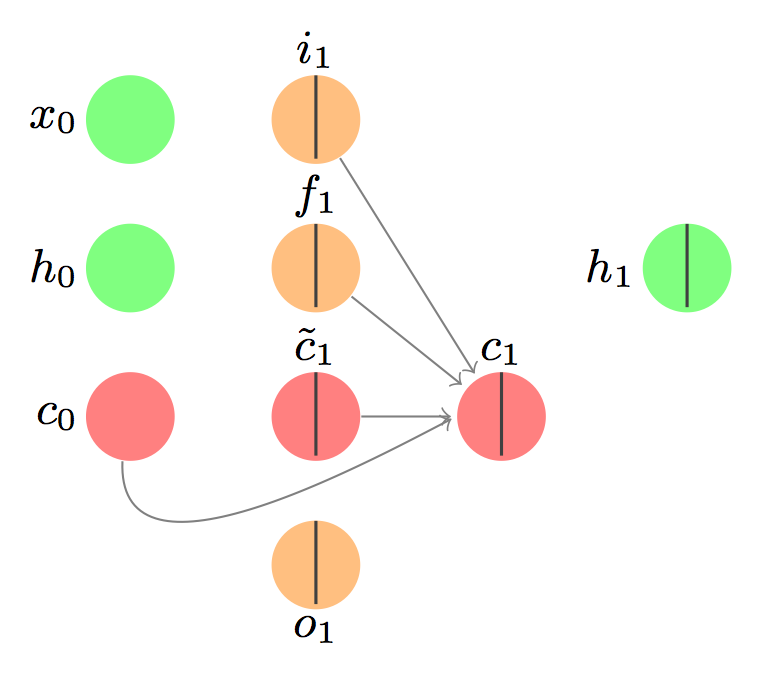
f_1 = \sigma({\rm{net}}_{f1}) = 1/(1 + exp(-0.005)) = 0.5012

Again, a stochastic decision could be made here as to whether the previous information should be forgotten (value 0) or allowed through (value 1). For the purposes of this example, let’s assume the value is 1.

### The memory cell

This is the best part! We combine the new information from the input gate and remove the information we’re forgetting according to the forget gate.

The picture is:



and the update looks like this:

c_1 = i_1 \circ \tilde{c}_1 + f_1 \circ c_0

That’s a new symbol! We need an aside.

The Hadamard product is an element-wise product. If we have a vector a_1 = [1, 2, 3] and a vector b_1 = [9, 10, 11], then the Hadamard product would be c_1 = a_1 \circ b_1 = [(1)(9), (2)(10), (3)(11)] = [9, 20, 33].

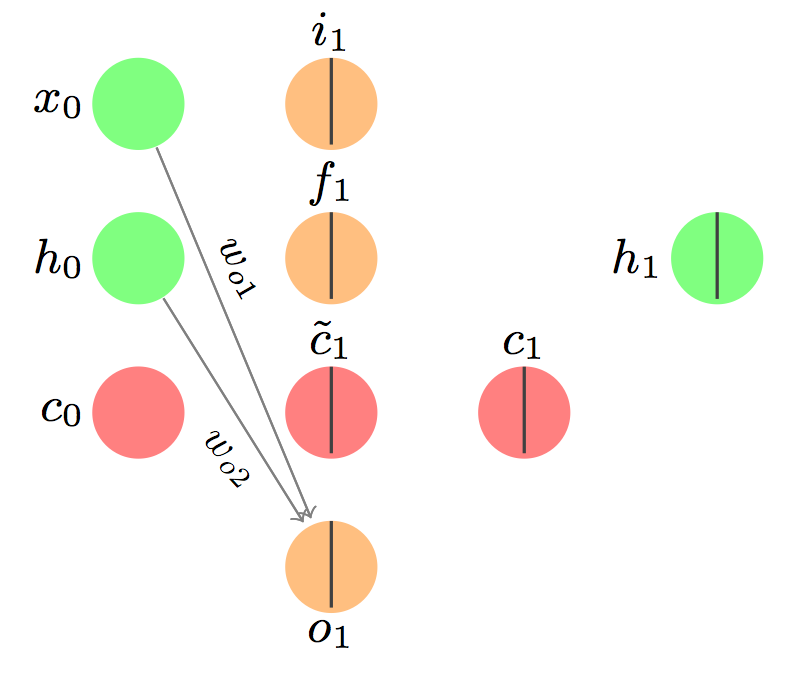
#### END ASIDE

c_1 = 1 \circ 0.0798 + 1 \circ 0 = 0.0798

Now that we’ve updated the memory state (another name for the memory cell), we have to think about what we want to output.

### The output gate.

Here’s the image to think of:



By now you should be thoroughly bored with these equations:

{\rm{net}}_{o1} = w_{o1}x_1 + w_{o2}h_0 + b_o

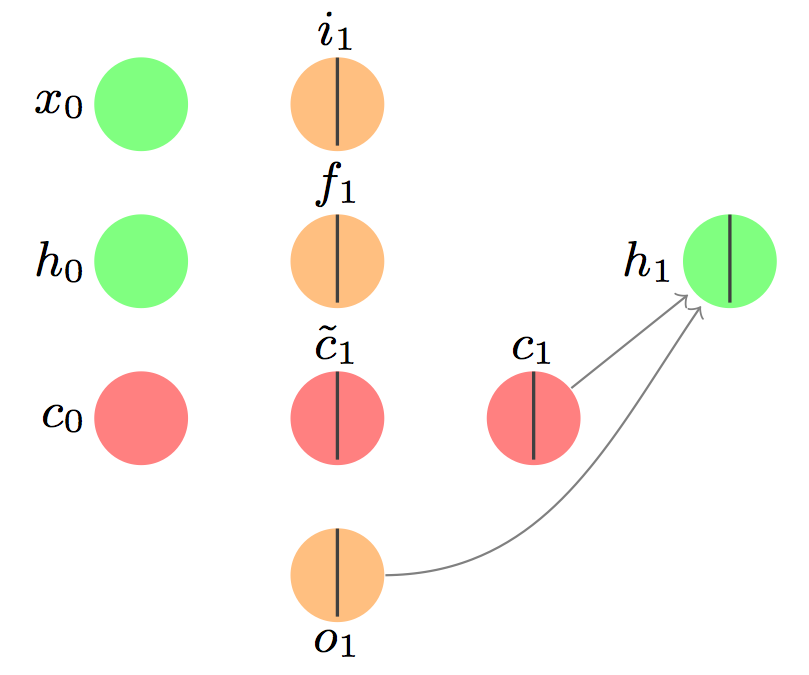
o_1 = \sigma({\rm{net}}_{o1}) = 1/(1 + exp(-{\rm{net}}_{o1} ))

{\rm{net}}_{o1} = 0.02(0.1) + 0.04(0) + 0.001 = 0.003

o_1 = \sigma({\rm{net}}_{o1}) = 1/(1 + exp(-0.003)) = 0.5007

And we’ll make a stochastic decision as to whether we pass this output along. For the purposes of this example, let’s assume the stochastic decision results in a 1.

### The hidden layer (hidden state)



I bet you were wondering when we’d get to this. The hidden layer is separate from the memory cell, but very related. I like to think of it as the part of the memory cell that we want to ensure persists. Here’s how we do it:

h_1 = o_1 \circ {\rm{tanh}}(c_1)  
h_1 = 1 \circ 0.0796 = 0.0796 (yes, I’m rounding. I’ve been doing that a lot.)

See what we did there? The output gate decides whether the signal from the memory cell gets sent forward as part of the input to the next LSTM cell.

### The second LSTM cell

We’ll assume the weights are shared across LSTM cells. The equations are exactly the same, but now we use x_1 where before we used x_0 and h_1 where we used h_0 and c_1 where we used c_0, etc. Let’s say we have input x_1 =0.2, and target scalar value y = 0.08. Here are all of the familiar computations written out, and final answers given (assuming all stochastic gate decisions result in the signal being propagated forward, and 0 information forgotten):

Input gate:

{\rm{net}}_{i2} = w_{i1}x_2 + w_{i2}h_1 + b_i  
i_2 = \sigma({\rm{net}}_{i2}) = 1/(1 + exp(-{\rm{net}}_{i2} )) = 0.52875  
{\rm{net}}_{c2} = w_{c1}x_2 + w_{c2}h_1 + b_c  
\tilde{c}_2 = \sigma({\rm{net}}_{c2}) = 1/(1 + exp(-{\rm{net}}_{c2} )) = 0.11768

Forget gate:

{\rm{net}}_{f2} = w_{f1}x_2 + w_{f2}h_1 + b_f  
f_2 = \sigma({\rm{net}}_{f1}) = 1/(1 + exp(-{\rm{net}}_{f2} )) = 0.50231

Memory cell:

c_2 = i_2 \circ \tilde{c}_2 + f_2 \circ c_1 = 0.61999

Output gate:  
{\rm{net}}_{o2} = w_{o1}x_1 + w_{o2}h_0 + b_o  
o_2 = \sigma({\rm{net}}_{o2}) = 1/(1 + exp(-{\rm{net}}_{o2} )) = 0.50145

Hidden state:  
h_2 = o_2 \circ {\rm{tanh}}(c_2) = 0.5511

Okay, now we’ve reached the end of our sequence, so we only need to use the hidden state to consider the entire memory of our LSTM.

The calculation:

\hat{y} = w_y h_2 + b_y = 0.6(0.5511) + 0.025 = 0.335566

That value \hat{y} is our final output.

But wait, weren’t we aiming for 0.08? We need to make some changes to our model. To do that, we’ll calculate the error and backpropagate the signal to update our weights.

The error (mean squared error, or MSE, but with only one value so ‘mean’ is irrelevant):

E = \frac{1}{2}(y - \hat{y})^2 = 0.5*(0.335566-0.08)^2 = 0.03266

The first step is to calculate the gradient of the error with respect to the output:

\frac{\partial E}{\partial o_t} = \frac{\partial E}{\partial h_t} \frac{\partial h_t}{\partial o_t} = (y - \hat{y})(-w_y){\rm{tanh}}(c_t)

We can see the dependency on the hidden state by expanding \hat{y}:

\frac{\partial E}{\partial o_t} = (y - [w_y(h_t) + b_y])(-w_y){\rm{tanh}}(c_t) = \delta_{o_t}

I’ll use \delta_i to refer to the partial derivative of the error with respect to i, similar to that blog post.

Now we need to differentiate through the hidden state to get to the next part. Alternatively, we could differentiate through c_2 directly – that’s the second path the gradient can take.

\frac{\partial E}{\partial c_t} = \frac{\partial E}{\partial h_t} \frac{\partial h_t}{\partial c_t} = (y - [w_y(h_2) + b_y])(-w_y)(o_t)(1 - {\rm{tanh}}^2(c_t)) = \delta_{c_t}

Now we need to go through the input and forget gates.

The input gate:

\frac{\partial E}{\partial i_t} = \frac{\partial E}{\partial c_t} \frac{\partial c_t}{\partial i_t} = \delta_{c_t} \tilde{c_t} = \delta_{i_t}

The forget gate:

\frac{\partial E}{\partial f_t} = \frac{\partial E}{\partial c_t} \frac{\partial c_t}{\partial f_t} = \delta_{c_t} c_{t-1} = \delta_{f_t}

The proposal for the new memory state:

\frac{\partial E}{\partial \tilde{c_t}} = \frac{\partial E}{\partial c_t} \frac{\partial c_t}{\partial \tilde{c_t}} = \delta_{c_t} i_t = \delta_{a_t}

The previous cell state:

\frac{\partial E}{\partial c_t} = \frac{\partial E}{\partial c_t} \frac{\partial c_t}{\partial \tilde{c}_t} = \delta_{c_t} f_t = \delta_{c_{t-1}}

The input to the proposal:

\frac{\partial E}{\partial {\rm{net}}_{c_t}} = \frac{\partial E}{\partial c_t} \frac{\partial c_t}{\partial \tilde{c}_t} \frac{\partial \tilde{c}_t}{\partial {\rm{net}}_{c_t}} = \delta_{a_t} (1-tanh^2({\rm{net}}_{c_t})) = \delta_{\hat{a}_t}

The net input to the input gate:

\frac{\partial E}{\partial {\rm{net}}_{i_t}} = \frac{\partial E}{\partial c_t} \frac{\partial c_t}{\partial i_t} \frac{\partial i_t}{\partial {\rm{net}}_{i_t}} = \delta_{i_t} i_t(1 - i_t) = \delta_{\hat{i}_t}

because of the derivative of the sigmoid function

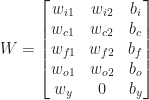
The net input to the forget gate:

\frac{\partial E}{\partial {\rm{net}}_{f_t}} = \frac{\partial E}{\partial c_t} \frac{\partial c_t}{\partial f_t} \frac{\partial f_t}{\partial {\rm{net}}_{f_t}} = \delta_{f_t} f_t(1 - f_t) = \delta_{\hat{f}_t}

The net input to the output gate:

\frac{\partial E}{\partial {\rm{net}}_{o_t}} = \frac{\partial E}{\partial c_t} \frac{\partial c_t}{\partial o_t} \frac{\partial o_t}{\partial {\rm{net}}_{o_t}} = \delta_{o_t} o_t(1 - o_t) = \delta_{\hat{o}_t}

Now we need to recall our definitions from way up top:



And let I_t be the total input at time t: [x_t, h_{t-1}, 1]^T.

Then we can define z_t = W I_t, and collect all of our ‘lowest’ derivatives:

\delta_{z_t} = [\delta_{\hat{i}_t}, \delta_{\hat{a}_t}, \delta_{\hat{f}_t}, \delta_{\hat{o}_t}]

Then our last derivatives are:

\frac{\partial E}{\partial I_t} = W^T \delta_{z_t}

and

\frac{\partial E}{\partial W_t} = \delta_{z_t} X (I_t)^T.

And there you have it! Backpropagation through an LSTM.

**RESNEXT:-**

A diagram of a rectangular object

Description automatically generated with medium confidence

A white rectangular object with a black background

Description automatically generated

# Block 1

## 1 convolution

We are replicating the simplified operation for every layer on the paper.

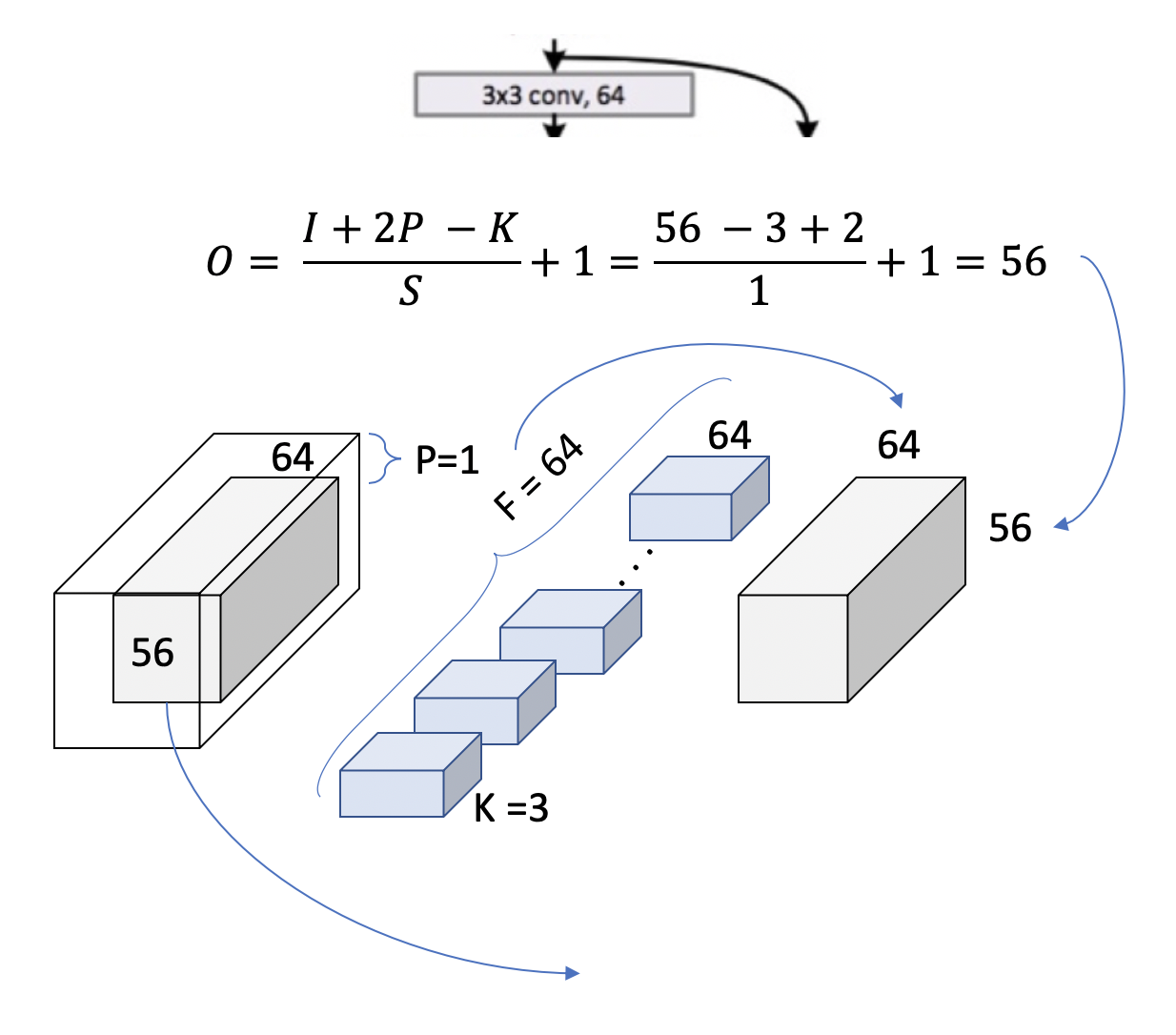
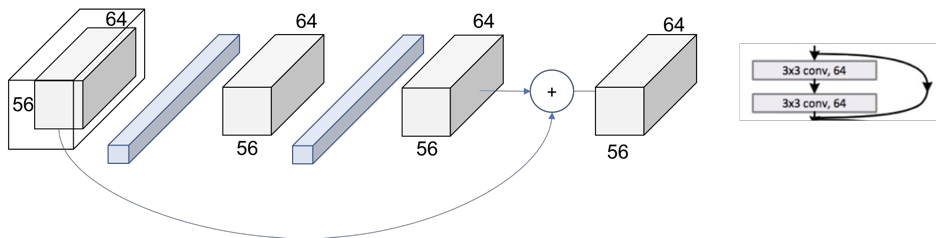


Figure 6. Layer 1, block 1, operation 1

We can double check now in the table from the paper we are using [3x3, 64] kernel and the output size is [56x56]. This is because a padding = 1 is used and a stride of also 1. Let’s see how this extends to an entire block, to cover the 2 [3x3, 64] that appears in the table.



Now, **we can completely read the whole cell of the table** (just recap we are in the 34 layers ResNext at Conv2\_x layer.

We can see how we have the **[3x3, 64] x 3 times within the layer**.

A close-up of a black background

Description automatically generated

Figure 8. Layer 1

# Patterns

# This means that the**down sampling of the volume though the network is achieved by increasing the stride instead of a pooling operation** like normally CNNs do.

We can also see another repeating pattern over the layers of the ResNet

A white box with blue dots

Description automatically generated

Figure 9. Layer2, Block 1, operation 1

The number of filters is duplicated in an attempt to preserve the time complexity for every operation (56\*64 = 28\*128).

A white box with blue dots and a black background

Description automatically generated

Figure 10. Projection Shortcut

The final picture looks then like in Figure 11 where now the 2 output volumes of each thread has the same size and can be added.

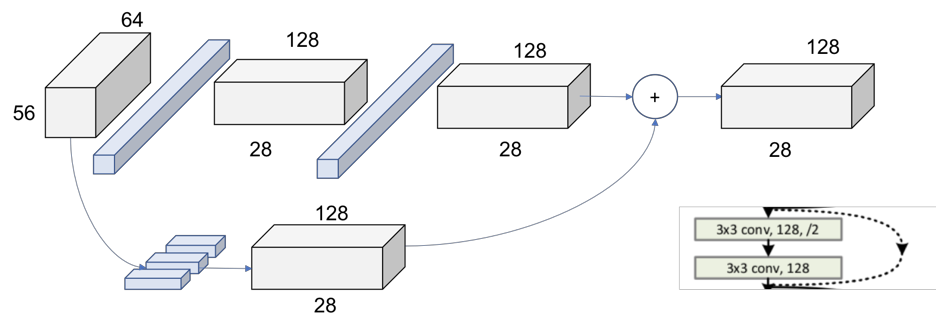


Figure 11. Layer 2, Block 1

In Figure 12 we can see the global picture of the entire second layer. The behavior is exactly the same for the following layers 3 and 4, changing only the dimensions of the incoming volumes.

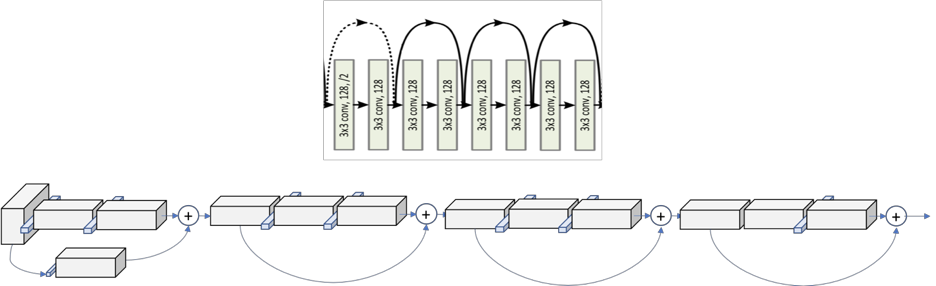


Figure 12.Layer 2

# Summary

