Digital Forensic

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

Yashraj Jain (205121107)

DEPARTMENT OF COMPUTER APPLICATIONS

NATIONAL INSTITUTE OF TECHNOLOGY,
TIRUCHIRAPPALLI 620015

DECEMBER 2023

BONAFIDE CERTIFICATE

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

Yashraj Jain (205121107)

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. U. Srinivasulu Reddy	**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. U. Srinivasulu Reddy**, Associate 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 **Project Evaluation Committee**, 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, propelled by advanced artificial intelligence (AI) and machine learning algorithms, has emerged as a formidable tool for crafting hyper-realistic fake videos, posing a multifaceted threat to trust, privacy, and security. This abstract delves into the pivotal domain of Deepfake Video Detection AI, a crucial field dedicated to formulating sophisticated algorithms and techniques aimed at identifying and mitigating the risks associated with deceptive content.

The research explores the intricate principles governing deepfake generation, shedding light on the intricate process of manipulating visual and auditory elements to fabricate convincing yet entirely synthetic videos. As deepfake technology evolves, so do the challenges in distinguishing genuine content from meticulously crafted fabrications. Current detection methodologies, discussed in this abstract, range from traditional forensic approaches to cutting-edge AI-based solutions, each with its strengths and limitations.

By comprehensively examining the landscape of deepfake video detection, this abstract aims to contribute to ongoing efforts geared toward safeguarding individuals, organizations, and societies from the insidious applications of this rapidly evolving technology. The insights provided herein emphasize the critical need for continuous research, collaborative initiatives, and ethical considerations in the development of AI solutions. Addressing the challenges posed by deepfake videos requires a holistic and interdisciplinary approach, incorporating technical advancements, legal frameworks, and ethical guidelines.

In conclusion, the abstract advocates for a proactive stance in combating the challenges posed by deepfake videos. By fostering collaboration between researchers, industry experts, and policymakers, it is possible to develop and deploy robust AI solutions that not only detect and mitigate deepfake threats but also adhere to ethical standards. This collaborative and forward-thinking approach is essential in ensuring the responsible and beneficial use of AI technologies in the face of evolving threats to trust, privacy, and security.

Table of Contents

Title Page no.

Table Of Figures

1. Introduction

1.1 Definition and History:

Deepfake technology is a word derived from the combination of the words "deep learning" and "fake" and emerged as a direct result in the mid-2010s due to major advances in deep learning and neural networks. The essence of deepfake technology is the use of complex algorithms to replace or create visual and audio content with real faces. The word “deep” indicates hope for deep learning techniques with neural networks that can learn complex patterns and features from large data sets.

The essence of deepfake technology lies in its ability to use artificial intelligence and machine learning to innovate. People seem to have a habit of combining misinformation with existing information. This combination of technologies allows the creation of content that is seemingly indistinguishable from a real visual or audio experience. This ability has major implications for everything from entertainment to more serious decisions like data breaches and privacy violations.

1.2 Evolution of Deepfake Technology:

In the embryo stage, deepfake technology found its place in the literature, especially in the entertainment industry. Early use cases include realistic animations and the creation of computer-generated characters, demonstrating the technology's ability to enhance the visual experience. But as deep learning algorithms have become more widespread, the development of deepfake technology has changed dramatically.

The technology has gone beyond its initial claims, leading to the creation of highly credible fake videos. These videos often show people saying or doing things they did not actually do, raising serious concerns about abuse. The development of deepfake technology has highlighted the importance of ethics and relationships, as well as the use of visual and audio information.

This development has led to a profound shift in the technology of the viewer, from a tool for creative expression to a potential impact on truth, reality and privacy. As technology continues to evolve, a critical understanding of its origin and evolution is needed to address the many challenges it poses. In the complex environment created by the development of deep fake technology, it has become important to balance innovation and ethics.

2. How DeepFake Works ?

2.1 Deep Learning Algorithm:

The main role of deep learning technology is based on the complex task of deep learning algorithm, with a special feature such as the use of Generative Adversarial Networks (GAN) and autoencoders. GANs are the main source of deep learning, which works through the interaction of two neural networks (generator and discriminator). While the creator creates synthetic content, the moderator verifies the authenticity of the content. This attack process repeatedly enhances the created content until it reaches a level of realism that is indistinguishable from the real thing.

Autoencoders are another type of deep learning algorithms that help encode and decode data. By compressing and restructuring input data into a latent representation, autoencoders help extract complex patterns necessary to create deeper meaning.

2.2 Data collection and training:

The origin of deepfake involves collecting and then training deep learning models by quickly processing data. To achieve a high level of accuracy, many different types of information are needed, including facial expressions, gestures, voice nuances, and other specific factors affecting the target person. This information forms the basis of deep learning algorithms to learn the complexity of objects' counterparts and their behavior.

During training, the deepfake model refines its parameters by iterating the dataset. The algorithm adjusts its weight and bias to accurately reproduce the learned features. This training process is important to ensure that the model expands its understanding of people's goals and makes it possible to adapt to different situations and strategies.

2.3 Creating fake content:

When the training is completed, the deepfake algorithm enters the content creation phase. Using the information obtained from the training data, the algorithm can combine features to create a video or audio recording. The combination of learning from faces, gestures, and noise creates content that appears more realistic to human observers. The deepfake content created is not just a copy, but a creative synthesis that displays new content. Thanks to this process, deepfake technology achieves its goal of creating content that blurs the line between reality and fiction.

3. Applications Of DeepFake Technology

3.1 Entertainment Industry:

In the entertainment industry, deep technology has become a two-way advantage. On the plus side, it offers new solutions for creating realistic computer-generated characters and scenes. Filmmakers and animators can use deep animation to enhance visual effects, bring creative creatures to life, or integrate CGI content into action movies. The app has the ability to change the way movies and TV series are uniquely designed to work and deliver great results in other ways.

However, the benefits of using technology deep into the entertainment field also come with a notable downside. There is concern that the players will be replaced by players. In addition to raising ethical questions about the future of art, this also reveals the accuracy and depth of energy absorption that human beings can compare to each other.

3.2 Privacy Policy:

The impact of technology extends far beyond entertainment into the arena of political control. The ability to create fake videos of politicians poses a huge risk to public opinion, the electoral process and international relations. Criminals can use well-established technology to fake speeches, interviews or events involving politicians, influence public opinion, debate and even influence elections.

The advancement of deep-based political control complicates the credibility of messages broadcast by digital channels. It demonstrates the need for a robust system to verify the authenticity of audiovisual content, especially in the context of highly political messages.

3.3 Security and Cybersecurity Risks:

The dark side of deepfake technology appears as fraud and cybersecurity in this field. Criminals can use fake accounts to commit further fraud, identity theft, and cyberattacks. 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.

4. 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

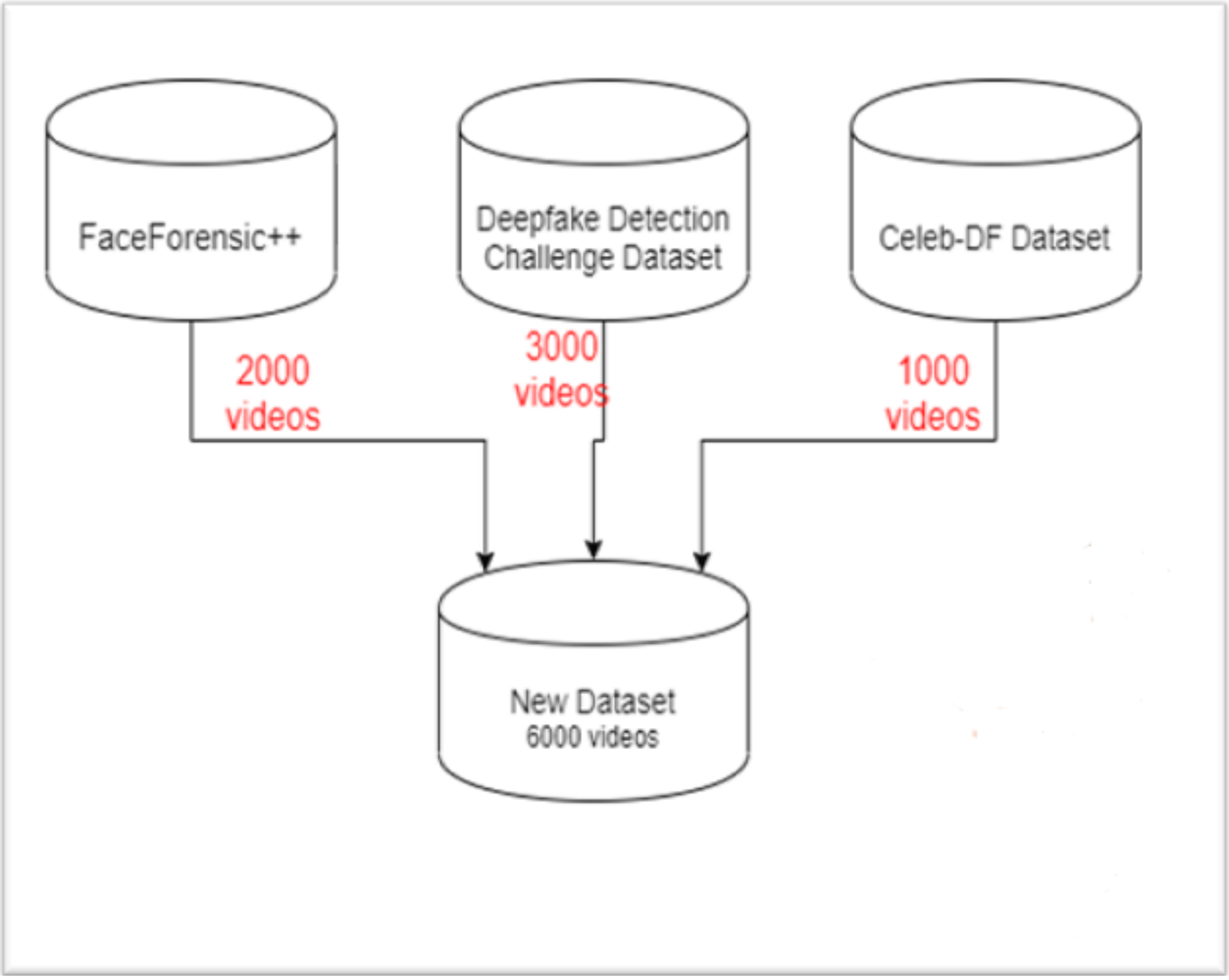


Fig 4.1 Dataset Collection

5. Challenges and Risks

5.1 Misinformation and Fake News:

The growth of deep content is causing serious problems in the form of misinformation and fake news. Because the technology has the ability to create trustworthy videos, there is an increased risk that criminals will use this ability to spread misinformation. Possible consequences include damage to trust in the media, manipulation of public opinion, and weakening of the democratic process. Detecting and combating misinformation based on deepfakes requires collaboration between technology developers and social media training programs, empowering the same people with the ability to see truth from context.

5.2 Privacy issues:

Deepfakes cause privacy issues, mostly caused by unauthorized control of personal portraits and personal information. Technology can create videos of people attending events they didn't attend or saying things they didn't say. This violates the individual's right to control their own image and interests and raises questions about the moral and legal nature of consent. As deepfakes become necessary, criminals can use personal information for a variety of purposes, including libel and slander, underscoring the urgent need for privacy and legal protection.

5.3 Law and Ethics:

Deepfake technology is developing faster than legal systems and ethical systems can be developed. This creates a difficult situation of uncertainty regarding business management, agreement, responsibility and enforcement of responsibility. Ethical considerations have become important when considering issues surrounding creative expression, the right to control digital media and possible interference in the relationship between exports. The balance between technological development and the protection of civil rights requires that legislators, experts, and technology developers work together to create processes that prevent abuse while promoting new roles.

6. Methodologies

6.1 LSTM :

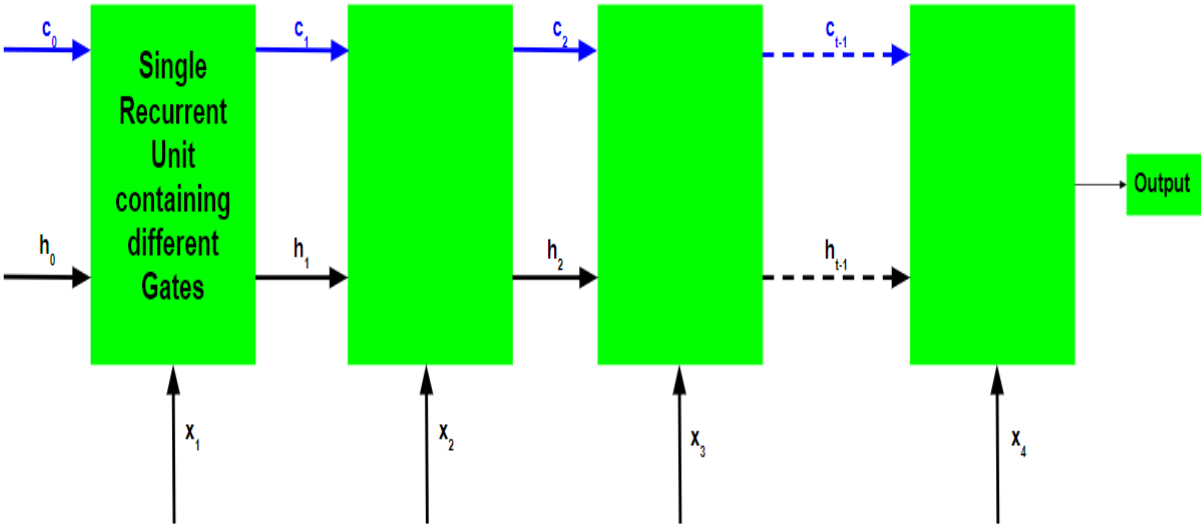


Fig 6.1.1 LSTM Implementation

NUMERICAL EXPALINATION: -

This is an LSTM with two cells:

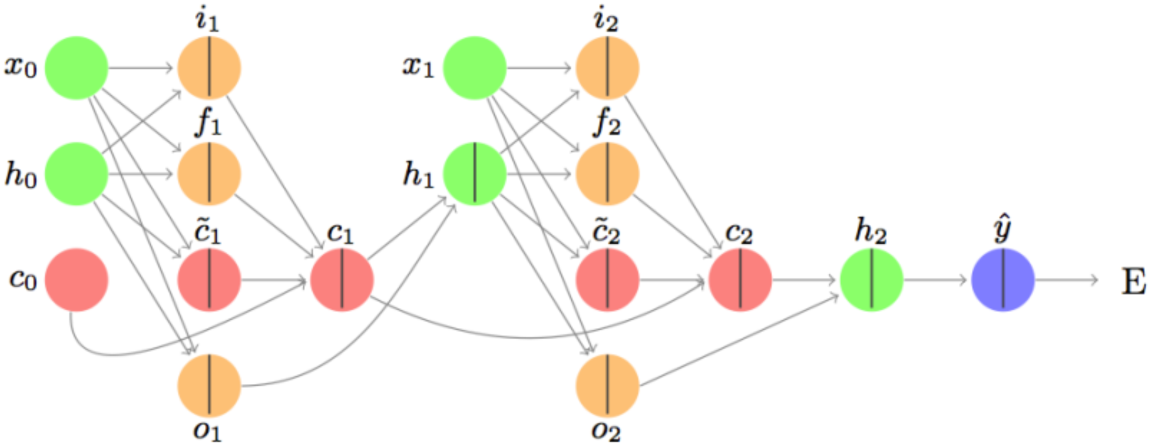


Fig 6.1.2 Two cells LSTM

Here is the first cell:

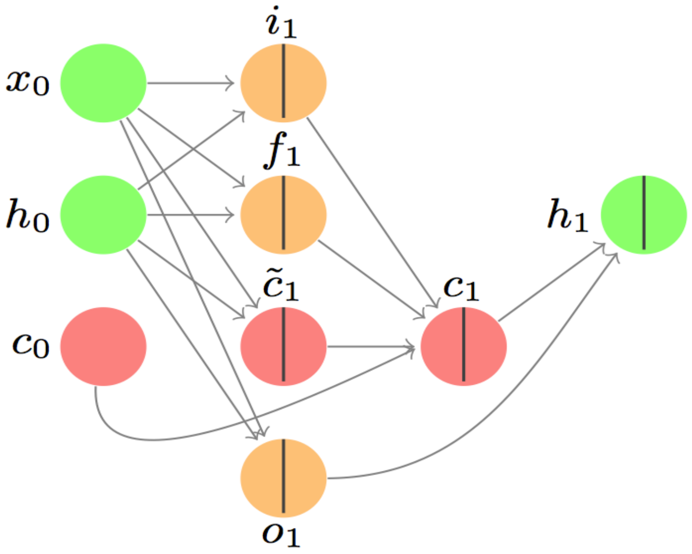


Fig 6.1.3 First Cell LSTM

Here is the second cell:

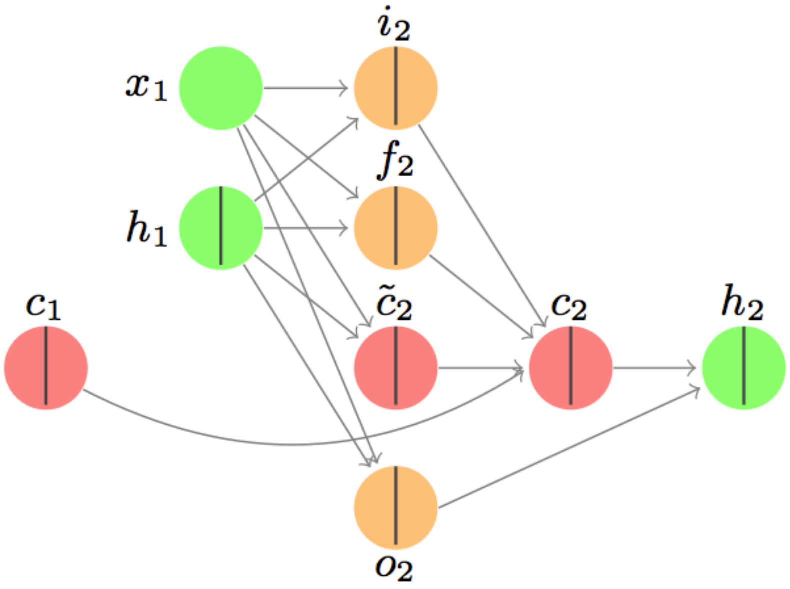


Fig 6.1.4 Second Cell LSTM

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.

$$W = \begin{bmatrix} w_{i1} & w_{i2} & b_i \\ w_{c1} & w_{c2} & b_c \\ w_{f1} & w_{f2} & b_f \\ w_{o1} & w_{o2} & b_o \\ w_y & 0 & b_y \end{bmatrix} = \begin{bmatrix} 0.5 & 0.25 & 0.01 \\ 0.3 & 0.4 & 0.05 \\ 0.03 & 0.06 & 0.002 \\ 0.02 & 0.04 & 0.001 \\ 0.6 & 0 & 0.025 \end{bmatrix}$$

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.

The input gate:

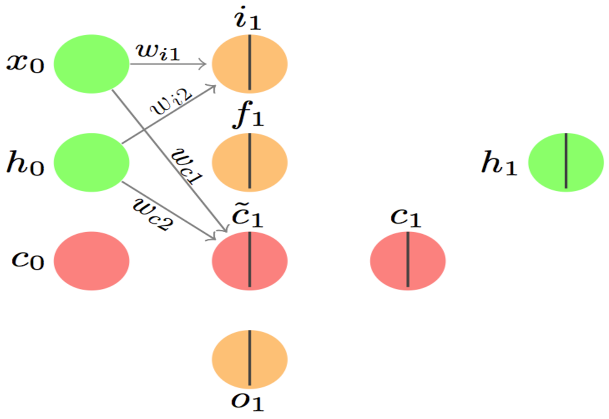


Fig 6.1.5 Input Gate

And the associated equations for the first part are:

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

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

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:

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

The full computation is:

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

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

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

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

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

The full computation is:

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

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

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 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.

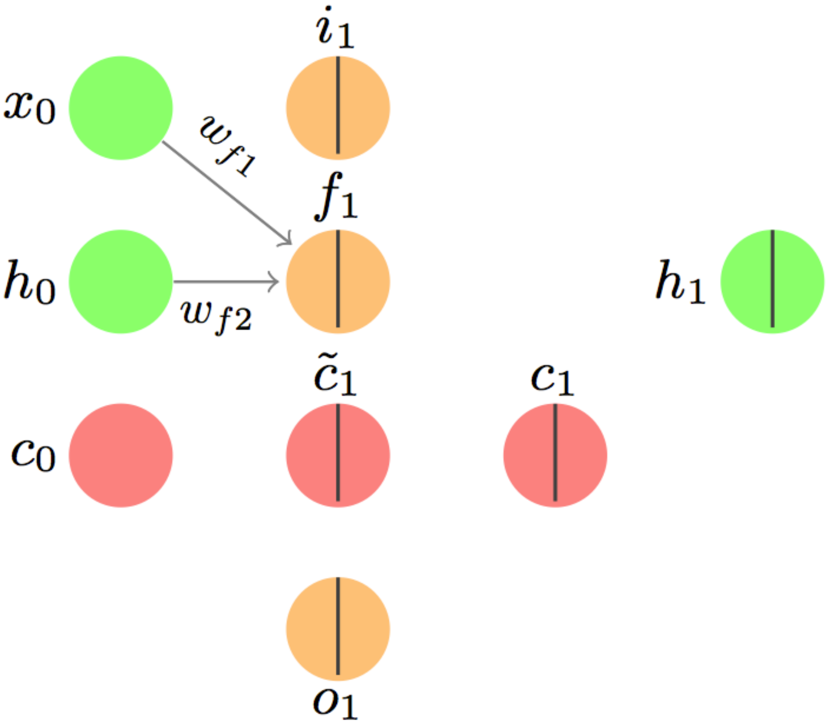


Fig 6.1.6 Forget gate

and takes similar input:

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

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

And the computation is also similar:

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

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

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:

We combine the new information from the input gate and remove the information we're forgetting according to the forget gate.

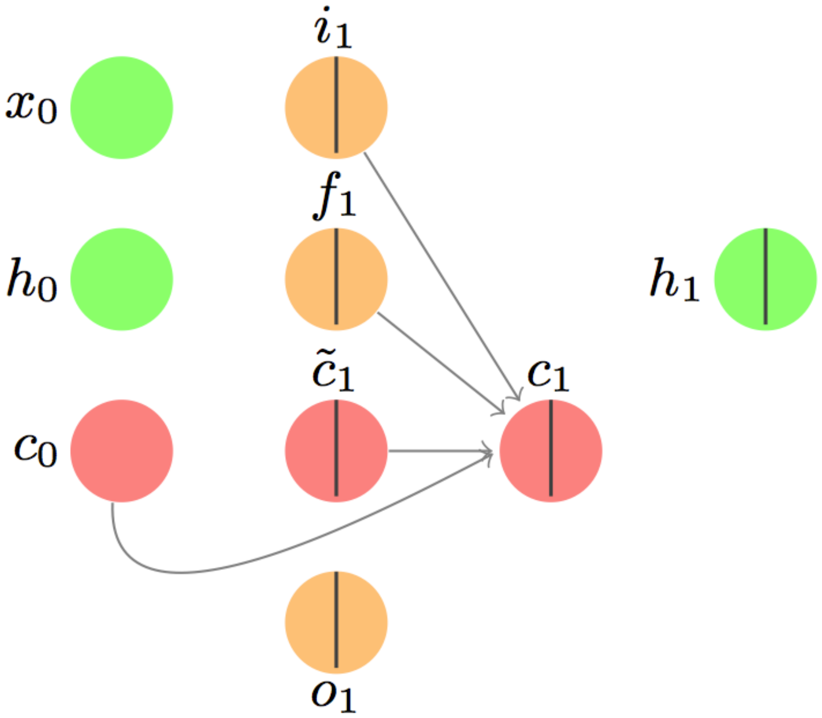


Fig 6.1.7 Memory Cell LSTM

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:

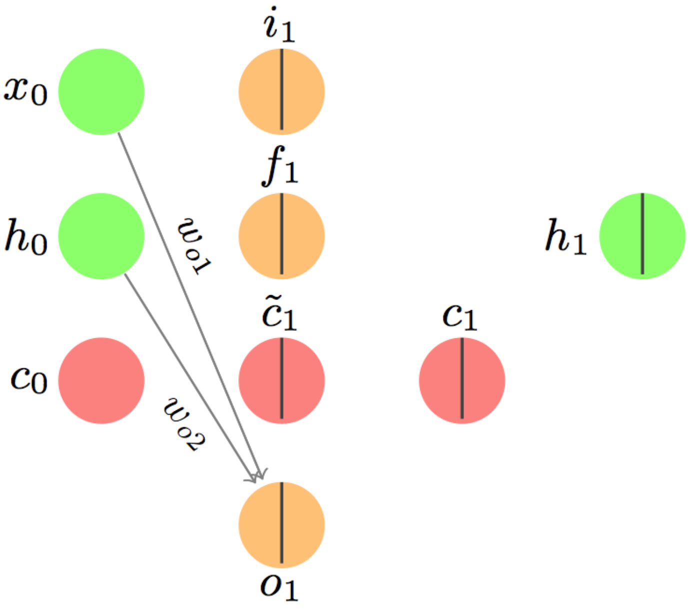


Fig 6.1.8 Output gate LSTM

By now you should be thoroughly bored with these equations:

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

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

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

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

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)

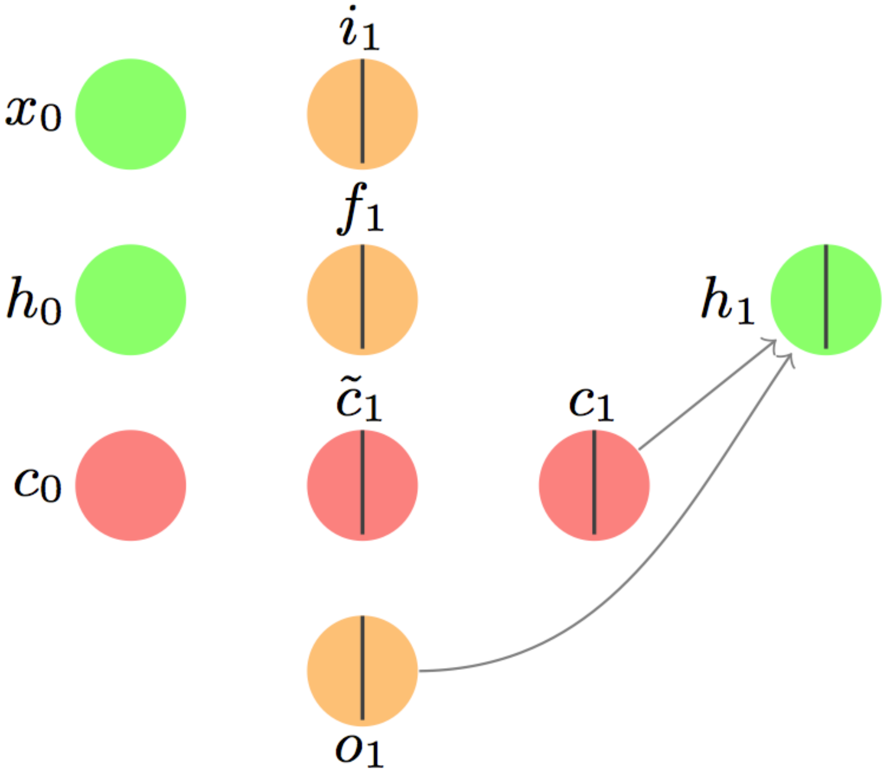


Fig 6.1.9 Hidden layer LSTM

The hidden layer is separate from the memory cell, but very related. Here's how we do it:

$$h_1 = o_1 \circ \tanh(c_1)$$
$$h_1 = 1 \circ 0.0796 = 0.0796$$

See 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:

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

Forget gate:

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

Memory cell:

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

Output gate:

$$\text{net}_{o2} = w_{o1}x_1 + w_{o2}h_0 + b_o$$
$$o_2 = \sigma(\text{net}_{o2}) = 1/(1 + exp(-\text{net}_{o2})) = 0.50145$$

Hidden state:

$$h_2 = o_2 \circ \tanh(c_2) = 0.5511$$

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:

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

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

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

I’ll use δ_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.

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

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:

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

The net input to the input gate:

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

because of the derivative of the sigmoid function

The net input to the forget gate:

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

The net input to the output gate:

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

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

$$W = \begin{bmatrix} w_{i1} & w_{i2} & b_i \\ w_{c1} & w_{c2} & b_c \\ w_{f1} & w_{f2} & b_f \\ w_{o1} & w_{o2} & b_o \\ w_y & 0 & b_y \end{bmatrix}$$

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.

6.2 ResNext:

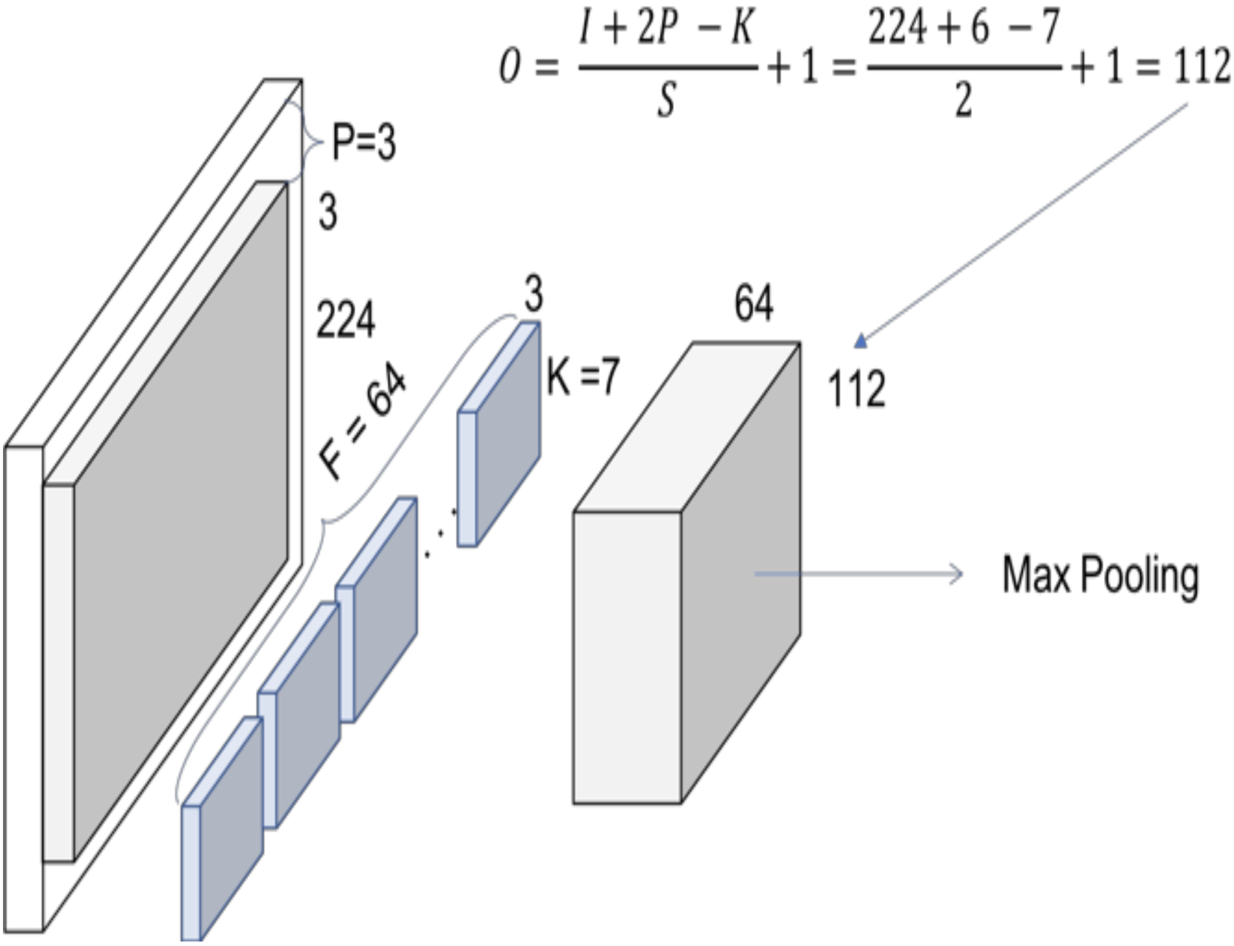

$O = \frac{I + 2P - K}{S} + 1 = \frac{224 + 6 - 7}{2} + 1 = 112$
P=3
3
224
F = 64
3
K =7
64
112
Max Pooling

Fig 6.2.1 ResNext Pooling

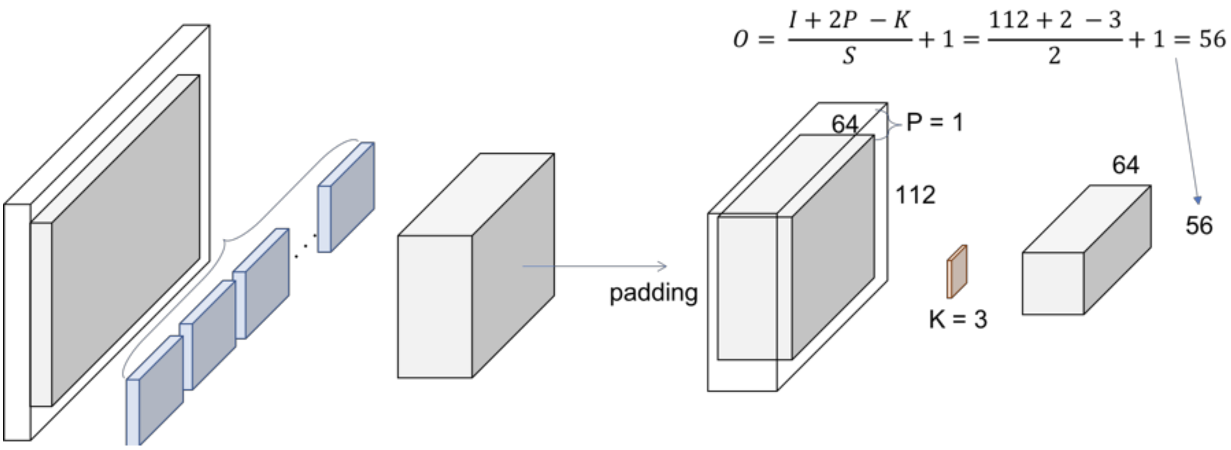
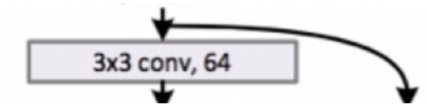

$O = \frac{I + 2P - K}{S} + 1 = \frac{112 + 2 - 3}{2} + 1 = 56$
64
P = 1
112
padding
K = 3
64
56

Fig 6.2.2 ResNext Padding

Block 1

1 convolution:

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



$$O = \frac{I + 2P \ - K}{S} + 1 = \frac{56 \ - 3 + 2}{1} + 1 = 56$$

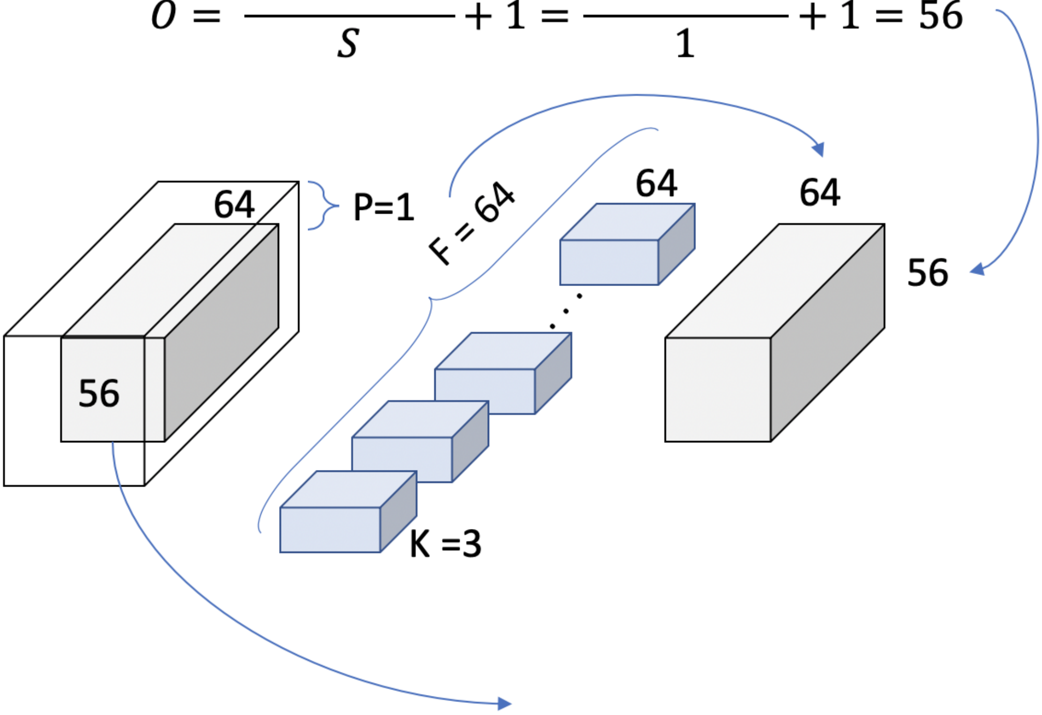


Fig 6.2.3 Layer 1, block 1, operation 1

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.

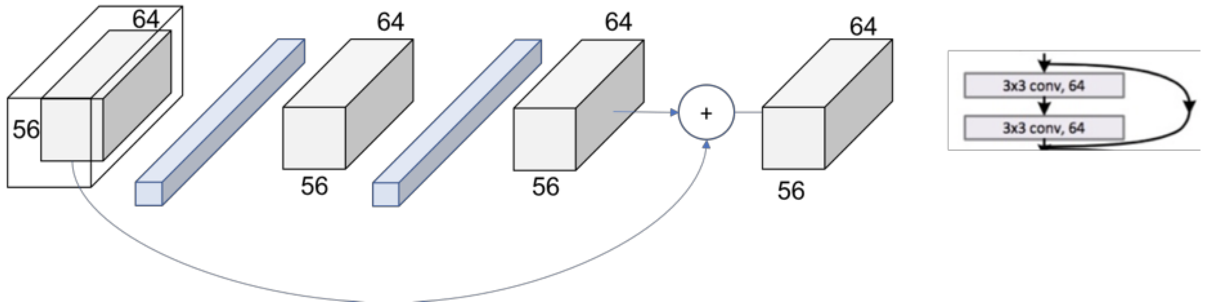


Fig 6.2.4 Layer 1, block 1, operation 2

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***.

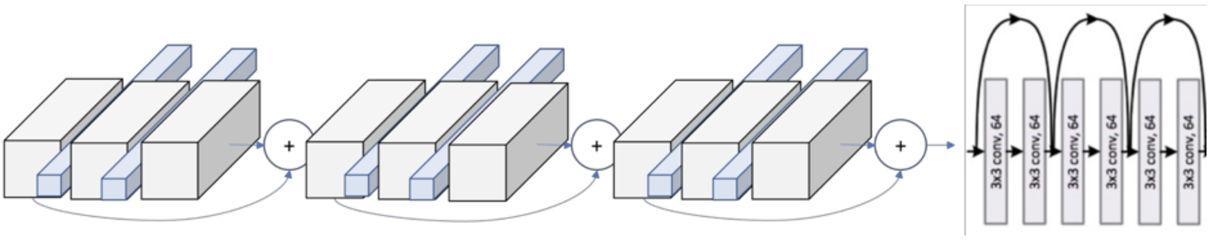
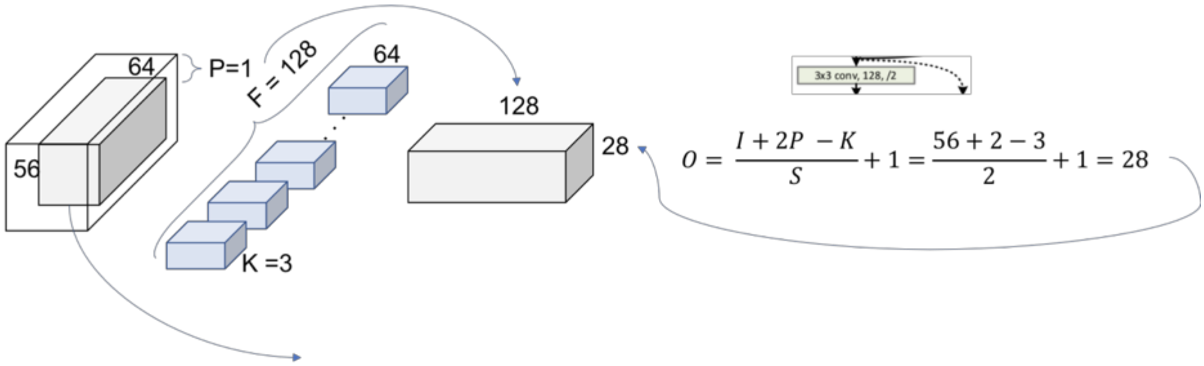


Fig 6.2.5 Layer 1

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



64
P=1
F = 128
64
56
128
28
K =3
3x3 conv, 128, /2
$O = \frac{I + 2P - K}{S} + 1 = \frac{56 + 2 - 3}{2} + 1 = 28$

Fig 6.2.6 Layer 2, 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*).

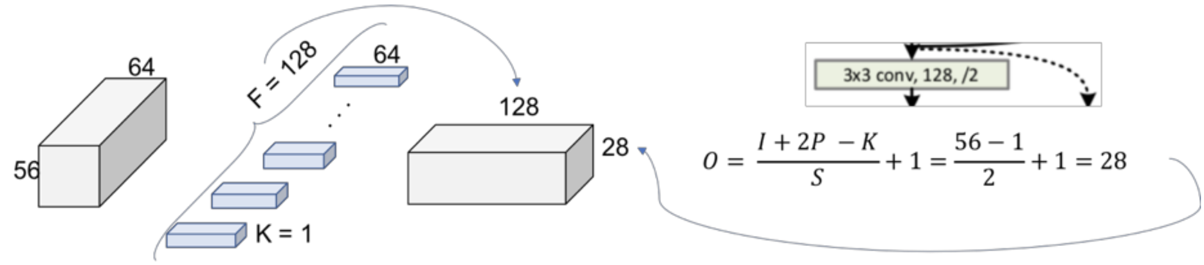


Fig 6.2.7 Projection Shortcut

The final picture looks then like in Figure 6.2.8 where now the 2 output volumes of each thread has the same size and can be added.

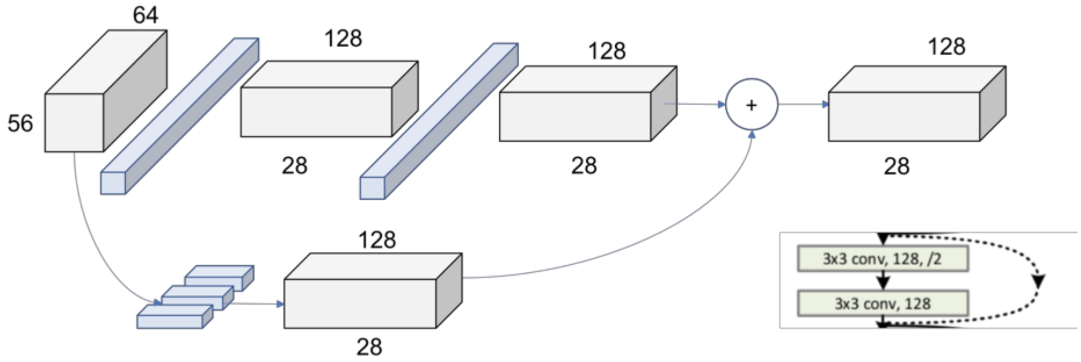


Fig 6.2.8 Layer 2, Block 1

In Figure 6.2.9 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.

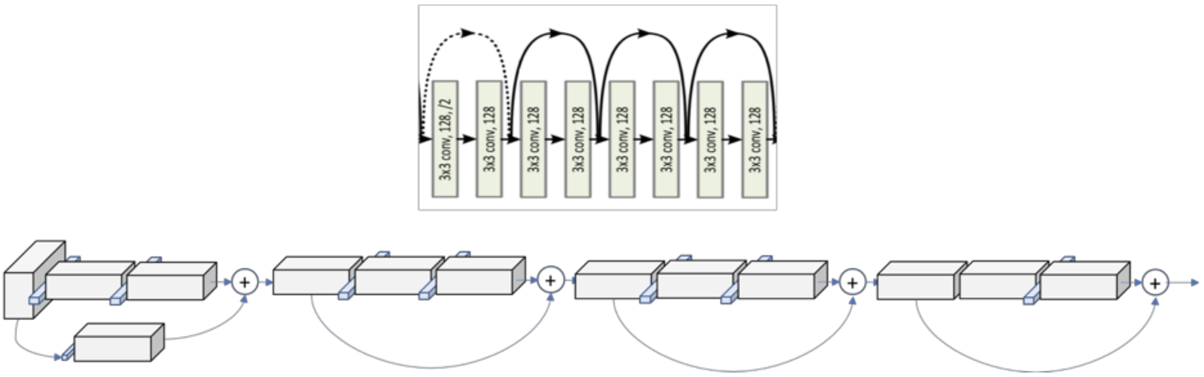


Fig 6.2.9 Layer 2

Summary:

Number of Layers	Number of Parameters
ResNet 18	11.174M
ResNet 34	21.282M
ResNet 50	23.521M
ResNet 101	42.513M
ResNet 152	58.157M

7. Project Implementation and Explanation

The project will follow these steps:

1. **Data Collection:**

    i) Identify Sources: Identify and gather videos from diverse sources, including social media, news outlets, and online video platforms. Ensure a balanced representation of real and deepfake videos to create a comprehensive dataset.

    ii) Annotation: Annotate the collected videos to clearly mark which ones are real and which ones are deepfakes. Accurate annotations are crucial for training a reliable model.

    iii) Quality Control: Implement quality control measures to filter out low-quality or irrelevant videos. This ensures that the dataset is composed of relevant and representative examples.

2. **Data Pre-processing:**

    i) Format Standardization: Convert videos to a standardized format to maintain consistency. This may involve converting videos to a specific resolution, frame rate.

    ii) Frame Extraction: Extract frames from videos to create a frame-level dataset. This allows the model to analyse individual frames for subtle manipulations.

    iii) Feature Extraction: Utilize computer vision techniques to extract relevant features from frames, such as facial landmarks, color histograms, or motion vectors.

3. **Model Training:** Train the CNN model on the pre-processed data. The model will learn to identify patterns in the videos that are indicative of deepfake manipulation.

4. **Model Evaluation:**

    i) Test Set Selection: Reserve a separate dataset for evaluation that was not used during training to assess the model's generalization capability.

    ii) Performance Metrics: Use metrics such as precision, recall, F1 score, and receiver operating characteristic (ROC) curves to evaluate the model's performance in distinguishing between real and deepfake videos.

    iii) Confusion Matrix Analysis: Examine the confusion matrix to identify specific types of errors made by the model, such as false positives or false negatives.

    iv) Threshold Optimization: Adjust decision thresholds to balance precision and recall based on the specific requirements of the deepfake detection application.

7.1 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.

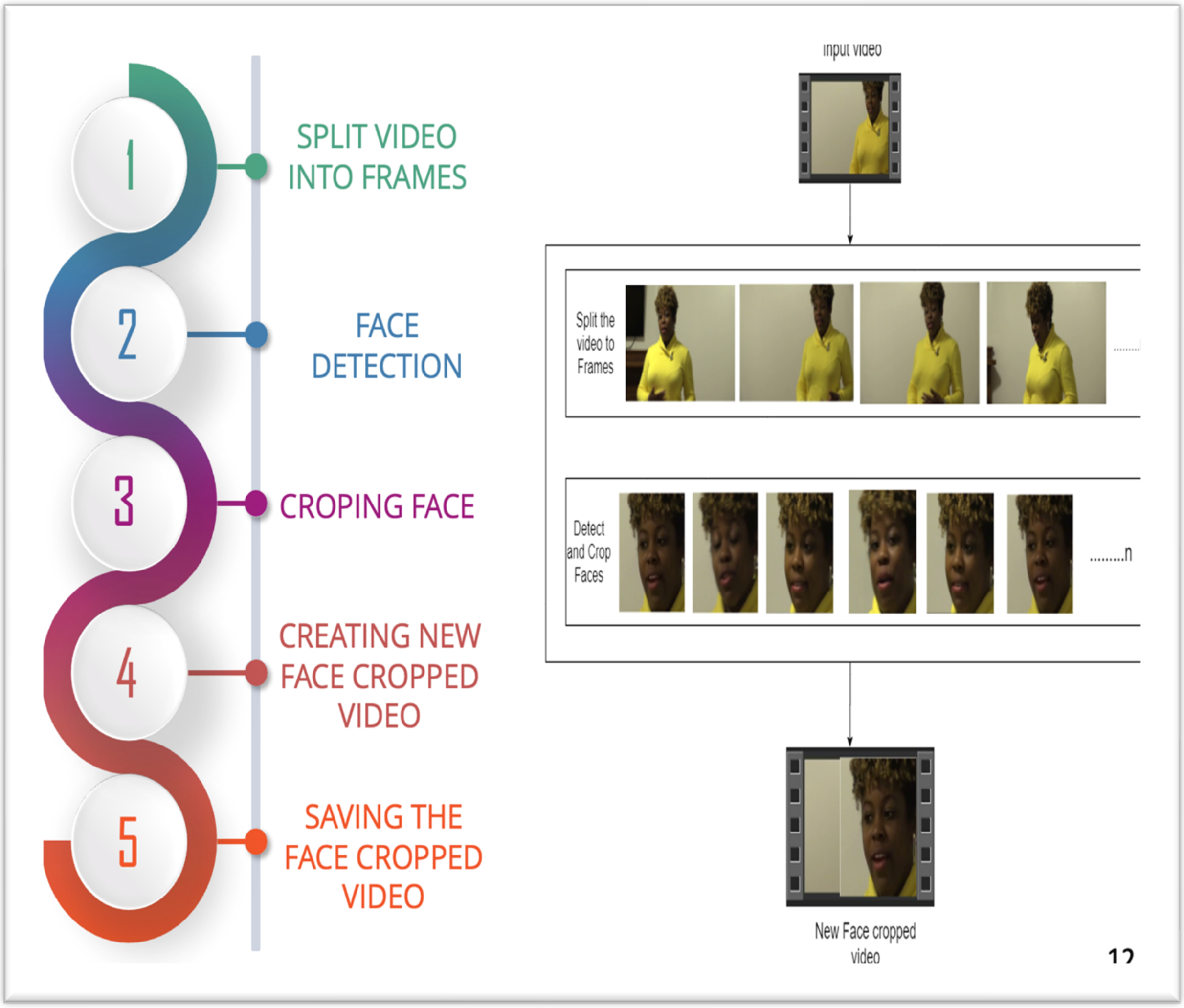
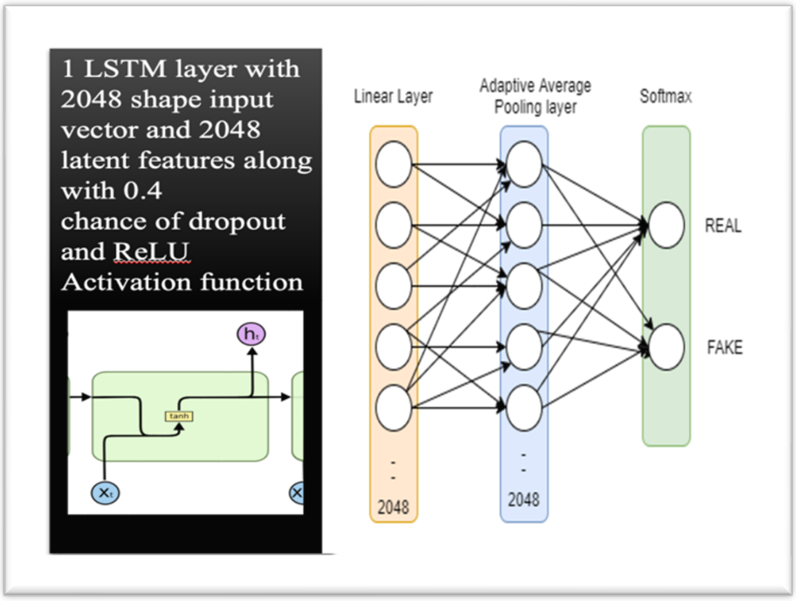


Fig 7.1 Preprocessing

7.2 Model Training and Evaluation: -

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 analyze temporal inconsistencies between frames introduced by the DeepFake creation process.



1 LSTM layer with 2048 shape input vector and 2048 latent features along with 0.4 chance of dropout and ReLU Activation function

Fig 7.2 LSTM Model Architecture

Stage	Output	ResNext-50 (32 x 4d)
Conv1	122 x 122	7 x 7, 64, stride 2
Conv2	56 x 56	3 x 3 max pool, stride 2 1 x 1, 128 3 x 3, 128 C= 32 1 x 1, 256
Conv3	28 x 28	1 x 1, 256 3 x 3, 256 C= 32 1 x 1, 512
Conv4	14 x 14	1 x 1, 512 3 x 3, 512 C= 32 1 x 1, 1024
Conv5	7 x 7	1 x 1, 1024 3 x 3, 1024 C= 32 1 x 1, 2048
	1x 1	Global average pool 1000 d fc, softmax

8. System Design

8.1 Architecture:

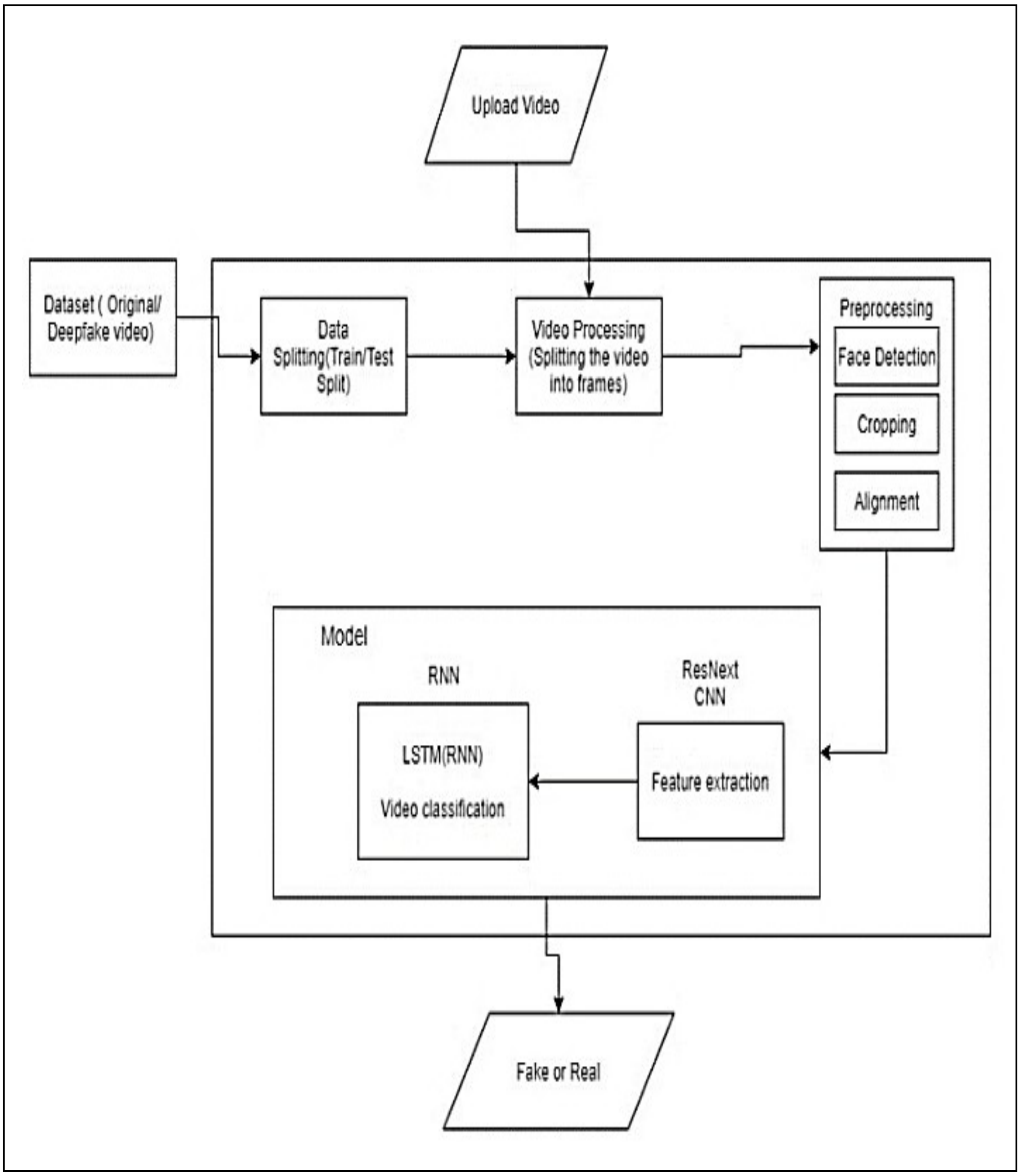


Fig 8.1 System Architecture

8.2 Flow Diagram:

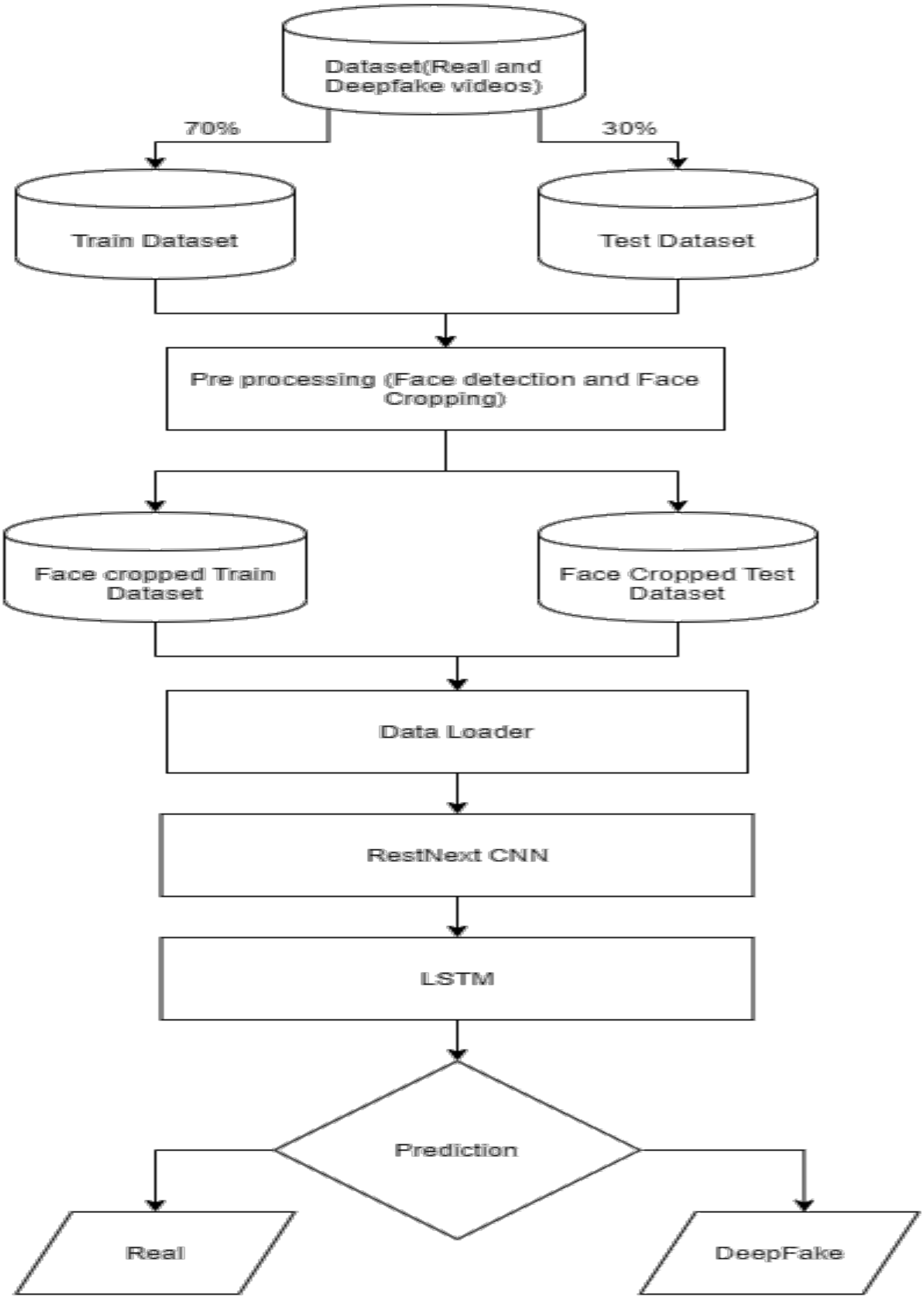

Dataset(Real and Deepfake videos)
70%
30%
Train Dataset
Test Dataset
Pre processing (Face detection and Face Cropping)
Face cropped Train Dataset
Face Cropped Test Dataset
Data Loader
RestNext CNN
LSTM
Prediction
Real
DeepFake

Fig 8.2 Flow diagram

9. Results

Model Name	Dataset	No. of videos	Sequence Length	Accuracy
3X3, conv,64	FaceForensic++	2000	20	90.95477387
3X3, conv,64			40	95.22613065
3X3, conv,64			60	97.48743719
3X3, conv,64			80	97.73366834
3X3, conv,64			100	97.76180905
6X6, conv,64	Our Dataset	6000	10	84.662519
6X6, conv,64			20	87.79160186
6X6, conv,64			40	89.34811819
6X6, conv,64			60	91.59097978
6X6, conv,64			80	92.49818558
6X6, conv,64			100	92.10883877

10. Conclusion

Confusion Matrix 1 :

	Predicted Yes	Predicted No	
Actual Yes	True Positive = 967	False Positive = 49	
Actual No	True Negative = 34	False Negative = 950	
			92.68

Accuracy Of the Model [3x3, 64] x 3 Is:- 92.68

Confusion Matrix 2:

	Predicted Yes	Predicted No	
Actual Yes	True Positive = 2530	False Positive = 342	
Actual No	True Negative = 278	False Negative = 2850	
			89.52

Accuracy Of the Model [6x6, 64] x 3 Is:- 89.52

11. Comparison

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.

12. Future Scope

12.1 Enhanced Audio Detection:

1) Integration of Audio Analysis: Extend the deepfake detection model to incorporate advanced audio analysis techniques. This can involve identifying anomalies in voice patterns, detecting unnatural pauses or artifacts in audio streams, and analyzing the consistency between audio and visual elements to create a more comprehensive deepfake detection system.
2) Multimodal Fusion: Explore the synergy between visual and audio cues by implementing multimodal fusion techniques. Combining information from both video and audio sources can enhance the model's accuracy and reliability in identifying sophisticated deepfake content.

12.2 Scaling Audio Detection:

1) Large-Scale Audio Dataset: Collect and curate a diverse dataset of real and manipulated audio files to facilitate the training of a robust audio detection model. Ensure that the dataset encompasses various accents, languages, and audio recording conditions to enhance the model's generalization capability.
2) Transfer Learning for Audio: Investigate the potential of leveraging pre-trained models on large audio datasets, enabling the model to learn generic features before fine-tuning for deepfake detection. Transfer learning can expedite the training process and improve performance, especially in scenarios with limited labeled audio data.

12.3 Handling Unstructured Video Files:

1) Video File Format Compatibility: Extend the model's capability to handle a wide range of video file formats and codecs, ensuring compatibility with diverse sources and applications. This involves adapting preprocessing techniques to accommodate unstructured video files commonly found on various platforms.
2) Dynamic Frame Rate Support: Develop mechanisms to handle videos with varying frame rates, resolutions, and compression levels. This adaptability will enable the model to effectively analyze unstructured video content encountered in real-world scenarios, including content from live streams and mobile devices.

12.4 Real-Time Deployment:

1) Optimization for Speed and Resource Efficiency: Investigate techniques for optimizing the deepfake detection model to achieve real-time or near-real-time processing speeds. This is crucial for deploying the model in applications where timely identification of deepfake content is paramount, such as live video streaming or social media moderation.
2) Edge Computing Integration: Explore the feasibility of deploying the deepfake detection model on edge devices, allowing for decentralized processing. This approach can enhance

privacy, reduce latency, and make the model more accessible in scenarios with limited internet connectivity.

12.5 Continuous Model Improvement:

1) Active Learning Strategies: Implement active learning strategies to continuously improve the model's performance over time. By identifying and prioritizing challenging examples for human verification, the model can adapt to emerging deepfake techniques and enhance its overall efficacy.
2) Feedback Mechanism: Establish a feedback loop that incorporates user feedback and manual verification into the training pipeline. This iterative process ensures that the model evolves to address new challenges and maintains high accuracy in detecting evolving deepfake methods.
3) By addressing these future scope areas, the deepfake detection model can evolve into a more comprehensive and adaptive solution capable of handling audio manipulation, diverse video formats, and real-time deployment scenarios.

13. REFERENCES

- Reference Research Paper:- https://ieeexplore.ieee.org/document/10083584
- Data-Set:- https://paperswithcode.com/task/deepfake-detection
- FaceForensics++:- https://paperswithcode.com/paper/faceforensics-learning-to-detect-manipulated
- LSTM Definition:- https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm/
- LSTM Components Explanation:- https://medium.com/@aidangomez/let-s-do-this-f9b699de31d9
- LSTM Working Principle:- https://www.kaggle.com/code/kmkarakaya/lstm-understanding-the-number-of-parameters
- ResNext Working Principle:- https://towardsdatascience.com/review-resnext-1st-runner-up-of-ilsvrc-2016-image-classification-15d7f17b42ac
- Resnet And ResNext:- https://d2l.ai/chapter_convolutional-modern/resnet.html
- Enhancing Resnet:- https://medium.com/dataseries/enhancing-resnet-to-resnext-for-image-classification-3449f62a774c

Websites

1. **Analytic Vidhya:- https://www.analyticsvidhya.com/**
2. **Stackoverflow:- https://stackoverflow.com/**
3. **Kaggle:- https://www.kaggle.com/**
4. **Medium:- https://medium.com/**