## ACKNOWLEDGEMENT

## 

We are extremely thankful to our beloved Chairman and Founder of MRGI **Sri. Ch. Malla Reddy,** for providing necessary infrastructure facilities throughout the project work.

We express our sincere thanks to Director **Dr. A. Ramaswamy Reddy**, MREC (A) who took keen interest and encouraged us in every effort during the project work.

We owe our gratitude to **Dr.A. Ravindra**, Principal, MREC(A) for his encouragement to accomplish the project work successfully.

We express our heartfelt thanks to **Dr. O.Obulesu**, Professor and Head, Department of Computer Science and Engineering, for her kind attention and valuable guidance throughout the project work.

We are thankful to our Project Coordinator **Mr. Pattlola Srinivas,** Associate Professor of CSE for his valuable suggestions and guidance throughout the project work.

We are extremely thankful to our Project Guide **Dr. S Dhanalakshmi**, Professor for her constant guidance and support to complete the project work.

We also thank all the teaching and non-teaching staff of Computer Science and Engineering Department for their cooperation during the project work

Mr. Mithilesh(16J41A05C8)

Mr. Ch. Nag Kamal(16J41A05D2)

Mr. G. Uday Kumar(16J41A05E5)

Mr. V. Satya Kumar(16J41A05H9)

Mr. S. Rakesh(17J45A0505)

**ABSTRACT**

We propose a deep learning method for single image super-resolution (SR). Our method directly learns an end-to-end mapping between the low/high-resolution images. The mapping is represented as a deep convolutional neural network (CNN)that takes the low-resolution image as the input and outputs the high-resolution one. We further show that traditional sparse-coding-based SR methods can also be viewed as a deep convolutional network. But unlike traditional methods that handle each component separately, our method jointly optimizes all layers. Our deep CNN has a lightweight structure, yet demonstrates state-of-the-art restoration quality, and achieves fast speed for practical on-line usage. We explore different network structures and parameter settings to achieve trade-offs between performance and speed. Moreover, we extend our network to cope with three color channels simultaneously, and show better overall reconstruction quality.

# LIST OF FIGURES

|  |  |  |  |
| --- | --- | --- | --- |
| **Figure No.** |  | **Name** | **Page No** |
|  |  |  |  |
| 3.2 |  | System flow | 9 |
| 5.1 |  | Approaches of single-image SR | 17 |
| 5.2 |  | Behaviour of different interpolation filters | 18 |
| 5.3 |  | Example-based super-resolution | 19 |
| 5.4 |  | Classification of Example-Based Super-Resolution | 19 |
| 6.1 |  | SRCNN Architecture | 21 |
| 6.2 |  | Patch extraction Non-linear Reconstruction and representation mapping | 24 |
| 6.3 |  | Example CNN sequence to classify handwritten digits | 25 |
| 6.4 |  | One Layer CNN | 28 |
| 6.5 |  | Three Layer CNN | 29 |
| 6.6 |  | Five Layer CNN | 29 |
| 9.1 |  | Use case diagram | 40 |
| 9.2 |  | Component diagram | 41 |
| 9.3 |  | Sequence diagram | 42 |
| 11.1 |  | Input images | 50 |
| 11.2 |  | Output images | 52 |

# ACRONYMS

|  |  |  |
| --- | --- | --- |
| **SR** | - | super-resolution |
| **SRCNN** | - | Super-Resolution Convolutional Neural Network |
| **SISR** | - | single image super resolution |
| **SR** | - | high-resolution |
| **PSNR** | - | peak signal-to-noise ratio |
| **RMSE** | - | root mean-square error |
| **LWT** | - | lifting wavelet transform |
| **SWT** | - | Stationary wavelet transforms |
| **HF** | - | high frequency |
| **ILWT** | - | inverse lifting wavelet transformation |
| **LSI** | - | linear space invariant |
| **AWGN** | - | additive white Gaussian noise |
| **HMRF** | - | Huber Markov random field |
| **SRS** | - | **software requirements specification** |
| **CNN** | - | Convolutional neural networks |
| **MLP** | - | The multi-layer perceptron |
| **SSIM** | - | structure similarity index method |

**INDEX**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CHAPTER NO** |  |  | **TITLE** | **PAGE**  **NO** |
|  |  |  |  |  |
| 1 |  |  | **INTRODUCTION** | 1 |
| 2 |  |  | **LITERARURE SURVEY** | 4 |
| 3 |  |  | **SYSTEM ANALYSIS** | 8 |
|  | 3.1 |  | Existing system | 8 |
|  | 3.2 |  | Proposed System | 9 |
|  | 3.3 |  | Feasibility Study | 10 |
|  | 3.4 |  | Feasibility Analysis | 10 |
|  |  | 3.4.1 | Operational Feasibility | 11 |
|  |  | 3.4.2 | Economic Feasibility | 11 |
|  |  | 3.4.3 | Technical Feasibility | 11 |
| 4 |  |  | **SOFTWARE REQUIREMENT ANALYSIS** | 12 |
|  | 4.1 |  | Software Requirement Specification | 12 |
|  | 4.2 |  | System Requirements | 14 |
|  |  | 4.2.1 | Software Requirements | 14 |
|  |  | 4.2.2 | Hardware Requirements | 14 |
| 5 |  |  | **CONVOLUTIONAL NEURAL NETWORKS FOR IMAGE SUPER-RESOLUTION** | 15 |
|  | 5.1 |  | Convolutional Neural Networks | 16 |
|  | 5.2 |  | Deep Learning for Image Restoration | 16 |
|  | 5.3 |  | Existing System for Super image Resolution | 17 |
|  |  | 5.3.1 | Interpolation-Based Super-Resolution | 17 |
|  |  | 5.3.2 | Reconstruction-Based Super-Resolution | 18 |
|  |  | 5.3.3 | Example-Based Super-Resolution | 18 |
|  |  | 5.3.4 | Parametric Vs Non-Parametric Methods | 20 |
|  |  | 5.3.5 | Internal Vs External Learning | 20 |
| 6 |  |  | **SYSTEM ARCHITECTURE** | 21 |
|  | 6.1 |  | Formulation | 21 |
|  |  | 6.1.1 | Path Extraction and Representation | 21 |
|  |  | 6.1.2 | Non-Linear Mapping | 22 |
|  |  | 6.1.3 | Reconstruction | 23 |
|  | 6.2 |  | Relationship Sparse-Coding Based Methods | 23 |
|  | 6.3 |  | Convolutional Neural Network (CNN) | 25 |
|  | 6.4 |  | Training | 26 |
|  | 6.5 |  | Training Data | 28 |
|  | 6.6 |  | Number of Layers | 28 |
| 7 |  |  | **PACKAGES** | 30 |
|  | 7.1 |  | Tensorflow | 30 |
|  | 7.2 |  | Keras | 31 |
|  | 7.3 |  | Skimage | 31 |
|  | 7.4 |  | Matplotlib | 32 |
|  | 7.5 |  | Opencv | 32 |
|  | 7.6 |  | Numpy | 32 |
|  | 7.7 |  | OS Module | 33 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
| 8 |  |  | **SOURCE CODE IMPLEMENTATION** | 34 |
|  | 8.1 |  | Importing Packages | 34 |
|  | 8.2 |  | Image Quality Metrics | 34 |
|  |  | 8.2.1 | Peak Signal to Noise Ratio (PSNR) | 35 |
|  |  | 8.2.2 | Mean Square Error (MSE) | 35 |
|  |  | 8.2.3 | Structure Similarity Index Method (SSIM) | 35 |
|  |  | 8.2.4 | Root Mean Square Error (RSME) | 36 |
|  | 8.3 |  | Preparing Images | 36 |
|  | 8.4 |  | Testing Low-Resolution Images | 37 |
|  | 8.5 |  | Building the Srcnn Model | 38 |
|  | 8.6 |  | Deploying the Srcnn | 38 |
| 9 |  |  | **UML DIAGRAMS** | 41 |
|  | 9.1 |  | Use case Diagrams | 41 |
|  | 9.2 |  | Component Diagram | 42 |
|  | 9.3 |  | Sequence Diagram | 43 |
| 10 |  |  | **SYSTEM TESTING** | 44 |
|  | 10.1 |  | Types of Tests | 44 |
|  |  | 10.1.1 | Unit Testing | 44 |
|  |  | 10.1.2 | Integration Testing | 44 |
|  |  | 10.1.3 | Functional Test | 45 |
|  |  | 10.1.4 | White Box testing | 45 |
|  |  | 10.1.5 | Black Box testing | 46 |
|  | 10.2 |  | Other Testing Methodologies | 46 |
|  |  | 10.2.1 | User Acceptance Testing | 46 |
|  |  | 10.2.2 | Output Testing | 46 |
| 11 |  |  | **APPLICATIONS** | 47 |
| 12 |  |  | **PROJECT RESULTS** | 50 |
|  | 12.1 |  | Input Images | 50 |
|  | 12.2 |  | Output Images | 51 |
| 13 |  |  | **CONCLUSION AND FUTURE ENHANCEMENTS** | 53 |
| 14 |  |  | **REFERENCES** | 55 |