



Automatic system with artificial vision for
grading multiple choice written tests for
educational institutions ruled by the LOEI

Ricardo Jumbo-Herrera and Milton Labanda-Jaramillo

EasyChair preprints are intended for rapid
dissemination of research results and are
integrated with the rest of EasyChair.

October 16, 2018

Sistema Automático con Visión Artificial para Calificar Pruebas Escritas de Base Estructurada para Instituciones Educativas Regidas por la LOEI

Abstract. This project develops a solution to the different problems that arise in the optical character recognition (OCR) in the field of education, specifically in the teaching area, focused on the recognition of characters in answer sheets, which are obtained from the application of written evaluations on a structured basis. To be implemented, Tesseract was used as the library for character recognition, for being robust, powerful and for the ability to be trained, also it uses Lepton as an OCR algorithm; the main problems are shown when recognizing the answer sheets, as well as a series of methods for correcting characters and techniques that allowed obtaining a better result. Finally, results are shown among a sample of 100 students of an educational institution in which the effectiveness of the system was tested.

Resumen. En este trabajo se desarrolla una solución informática orientada a los diferentes problemas que se presentan en el reconocimiento óptico de caracteres (OCR) en el ámbito de la educación, específicamente en el área de la docencia, enfocado al reconocimiento de caracteres en hojas de respuestas, las cuales se obtienen de la ejecución de evaluaciones escritas en base estructurada. Para su implementación, se utiliza Tesseract como librería para el reconocimiento de caracteres por ser robusta, potente y por la capacidad de poder ser entrenada, además utiliza Leptónica como algoritmo OCR; se muestran los principales problemas al momento de reconocer las hojas de respuestas, así como una serie de métodos de corrección de caracteres y técnicas que permitieron obtener un mejor resultado. Se muestra finalmente los resultados entre una población de 100 alumnos de una unidad educativa en la cual la efectividad del sistema fue puesta a prueba.

Keywords: OCR, Algoritmos, Leptónica, Técnicas OCRs.

1 Introducción

El Reconocimiento óptico de caracteres o OCR – por sus siglas en inglés, Optical Characters Recognition- es un proceso que tiene como objetivo digitalizar y reconocer los caracteres de un determinado alfabeto o simbología contenidos en diferentes fuentes de texto, como puede ser una hoja impresa por un procesador de palabras o texto escrito a

mano por un humano, donde este tiene el mayor nivel de dificultad para su reconocimiento por la diversa caligráfica de cada individuo; además de otros factores como el tamaño y la similitud entre símbolos.

El proceso de OCR viene ganando terreno en los últimos años, C. Isaza [1] utiliza esta tecnología en el ámbito empresarial dando solución al acceso a áreas restringidas de una empresa, mediante el OCR reconoce el DNI de los empleados y otorga los permisos correctos a esa persona. El OCR es muy utilizado para dar solución a problemas con códigos numéricos T. Arrighi [2], realizó un sistema que clasifica los productos de una empresa mediante un código numérico, al utilizar OCR reconoce este código numérico y clasifica automáticamente el producto, evitando la intervención humana en este proceso. B. Tejas [3] plantea como solución un sistema que permita reconocer la placa del vehículo y automáticamente extraiga los datos más importantes de ese vehículo, como el nombre del propietario, la dirección, la fecha, entre otros. El reconocimiento óptico de caracteres cada vez se está utilizando en el ámbito móvil, pues la facilidad que ahora se tiene con la cámara de los smartphone significa poder realizar OCR con nuestro teléfono como B. Vuong [4], que creó una aplicación móvil que utiliza el OCR para recuperar los datos del DNI y llenar un formulario automáticamente evitando que una persona digite esos datos.

La Ley Orgánica de Educación Intercultural (LOEI), establece que los exámenes trimestrales, supletorio, remedial y de gracia que se deben tomar en las instituciones públicas y privadas que se rigen a la LOEI deben ser de base estructurada que no contempla conceptos de memoria, si no que evalúan el conocimiento junto con las destrezas y actitudes del estudiante[5]. Con esta ley se vuelve vital tratar de automatizar procesos con respecto a las pruebas de base estructurada, con el fin de ayudar a la labor docente.

2 Software Usado

2.1 Reconocimiento Óptico de Carcateres (OCR)

El reconocimiento óptico de caracteres (OCR), es un proceso que permite digitalizar y reconocer caracteres por medio del uso de dispositivos fotoeléctricos y software, para poder lograrlo se ha hecho uso de varios métodos tradicionales como el algoritmo k-vecinos, las máquinas de soporte vectorial (SVM), KNN y redes neuronales. Para poder realizar el reconocimiento de caracteres se siguen ciertas etapas generales las cuales son [6, 7]:

- **Obtención de Imagen:** En esta etapa se puede obtener la imagen de dos formas, el modo “Offline” y el modo “Online”, en la primera se trabaja con imágenes digitales de un scanner, cámara o cualquiera otra fuente digital que contenga texto

para reconocer, en el método Online se utiliza datos que se obtienen en tiempo real de un dispositivo [6].

- **Segmentación:** El propósito de esta etapa es extraer cada carácter de la imagen, dividiendo la imagen en diferentes caracteres aislados que permite clasificarlos individualmente, esta etapa es imprescindible en cualquier sistema OCR [6].
- **Pre-Procesamiento:** Esta etapa se encarga de limpiar, normalizar y simplificar la imagen, dejándola lista para la siguiente etapa, algunos de los procesos más conocidos son [6]:
 - Auto-Cropping: En donde se remueve el espacio en blanco alrededor del carácter.
 - Normalización: Transforma las imágenes a un tamaño estándar.
 - Binarización: Este proceso se encarga de simplificar la imagen a un mapa de bits(bitmap) de 0 y 1s, los procesos se pueden observar en la Figura 2.

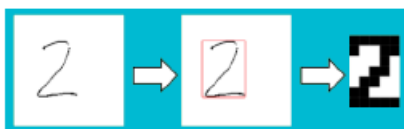


Fig. 1. Pre-procesamiento de Imagen.

- **Entrenamiento / Clasificación:** Esta última etapa consiste en asociar el conjunto de datos obtenidos en las anteriores etapas para asignar un carácter a una clase en particular, los métodos que se suele utilizar para lograr este fin puede ser K-NN, Template Matching, Support Vector (SVM) Machines y las Redes Neuronales [6].

2.2 Tesseract

Tesseract es un motor OCR libre. Fue desarrollado originalmente por Hewlett Packard como software propietario entre 1985 y 1995. Tras diez años sin ningún desarrollo, fue liberado como código abierto en el año 2005 por Hewlett Packard y la Universidad de Nevada, Las Vegas. Tesseract es desarrollado actualmente por Google y distribuido bajo la licencia Apache, versión 2.0. Tesseract está considerado como uno de los motores OCR libres con mayor precisión disponibles actualmente [8].

En 1995, Tesseract era uno de los tres mejores motores OCR en cuanto a precisión, además está disponible para Linux, Windows y Mac OS X, sin embargo, sólo ha sido probado por los desarrolladores en Windows y Ubuntu. Hasta la versión 2, Tesseract sólo podía aceptar como entrada imágenes de una sola columna en formato TIFF. En estas primeras versiones no se incluía análisis de patrones, y, por tanto, las imágenes con múltiples columnas o anotaciones producían resultados ilegibles. Desde la versión 3, Tesseract soporta el formato en el texto y el análisis del patrón de la página. A través de la biblioteca Leptonica, se consigue la compatibilidad con nuevos formatos de imagen, además, se puede detectar si el texto proporcional o monoespaciado. Tesseract

puede procesar más de 100 idiomas, y puede ser entrenado para funcionar con otros idiomas [8]

2.3 Stack MEAN

El stack Mean es utilizado para el desarrollo de aplicaciones web interactivas, mediante una poderosa combinación de cuatro tecnologías: MongoDB, Express.js, Angular.js y Node.js [9].

Todas estas tecnologías tienen en común la utilización de JavaScript como su principal característica, esto permite reducir el uso de varios lenguajes de programación y proporciona los componentes necesarios para el desarrollo del lado del cliente y del servidor, es de código abierto y gratuito.

Como se menciona anteriormente MEAN es la combinación de cuatro tecnologías [10], la primera es MongoDB la cual es una base de datos NoSQL que está orientada a documentos, para almacenar los datos utiliza el formato JSON (JavaScript Object Notation), Express.js es un framework para el servidor web Node.js, permite realizar las principales funciones del servidor e indexar las rutas para el API REST, después tenemos Angular.js que es un Framework para el desarrollo de aplicaciones de una sola página, controla las vistas del frontend mediante la arquitectura Modelo, Vista y Controlador (MVC) y finalmente Node.js que es un entorno del lado del servidor escrito en JavaScript y está basado en el motor Chrome V8 de Google.

2.4 Desarrollo Móvil

Android es el sistema operativo creado por Google para dispositivos móviles, está basado en Linux y desarrollado por Open Handset Alliance compuesto por más de 30 empresas de tecnología y compañías de telefonía móvil, como HTC, T-Mobile, Qualcomm, Motorola, Samsung, China Mobile y LG. Esta alianza busca establecer una plataforma de software para teléfonos móviles estandarizada y abierta. No hay límites entre las aplicaciones en la plataforma Android, por lo que se puede acceder a la funcionalidad del dispositivo a través de la API. Android es un entorno completo de tareas múltiples, por lo que las aplicaciones se pueden ejecutar en paralelo, de esta manera si una aplicación se está ejecutando en segundo plano, puede generar notificación [11].

Para la arquitectura de Android se la puede dividir en 4 capas, que va desde el núcleo del sistema operativo hasta la capa de aplicación como se muestra en la Figura 5.

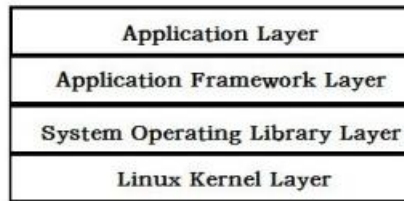


Fig. 2. Arquitectura de Android [11]

- **Application Layer:** la capa de aplicación se refiere a todos los programas que se escribieron en el lenguaje Java y se ejecutan en la máquina virtual, algunas aplicaciones ya vienen incluidas en el sistema operativo pues son consideradas principales.
- **Application Framework Layer:** Esta capa hace referencia a la API de las aplicaciones principales lanzadas por Google, los más beneficiados son los desarrolladores pues pueden aplicar estos frameworks para desarrollar sus propias aplicaciones, y de esta manera simplificar el diseño y arquitectura del programa, sin embargo para utilizar los framework Google establece los componentes que se deben utilizar para estandarizar las aplicaciones.
- **System Operating Library Layer:** Básicamente esta capa maneja todas las librerías que son necesarias para el sistema operativo, como algunas bibliotecas de C / C++ que mejoran el servicio, algunas de las librerías más importantes son: Bionic para C, Media Framework que permite soportar los formatos multimedia, SGL para gráficos en 2D, SSL se encarga de la comunicación mediante el protocolo TCP/IP, SQLite es una base de datos relacional.
- **Linux Kernel Layer:** Todos los servicios del sistema principal de Android están basados en el Kernel de Linux 2.5, como la seguridad, gestión de memoria, gestión de procesos, pila de protocolos de red. Se considera que el Kernel de Linux es una capa abstracta pues necesita de los controladores asociados con los controladores móviles para poder funcionar correctamente, los principales son: Driver de pantalla, Driver de teclado, Driver de Flash Memory, Driver de cámara, Driver de audio, Driver de Bluetooth, Driver del Wifi y la administración de energía.

3 Propuesta

La propuesta para el presente proyecto se centra en la implementación de un sistema web desarrollado con la tecnología stack Mean y una aplicación móvil desarrollada para dispositivos Android, para el desarrollo de la aplicación se hace uso del Ide Android Studio utilizando el lenguaje de programación Java, la aplicación permite la digitalización de las hojas de respuestas mediante la técnica de reconocimiento óptico de caracteres (OCR). Con la captura de una imagen digital desde la cámara del celular, la aplicación procesa esta imagen mediante el algoritmo leptonica incluida en la librería

Tesseract la cual devuelve un texto, este texto es procesado por un algoritmo que se encarga de realizar calificación de la evaluación y asignar la nota obtenida al estudiante.

4 Experimentos y Discusión

El sistema web se desarrolló utilizando el stack Mean que permite tener una plataforma con la facilidad para administrar las clases que tiene el docente, para cada clase se gestiona los alumnos y las evaluaciones. Para la calificación de las evaluaciones hacemos uso de una aplicación móvil, desarrollada bajo la plataforma Android, en la cual mediante la cámara se realiza la captura de la imagen para el posterior proceso de OCR. Como el objetivo de este proyecto es la de calificar pruebas escritas, hacemos uso de la librería Tesseract la cual es considerada como uno de los mejores OCRs del mercado.

Una vez que se tiene la imagen se aplica el proceso de OCR, con lo cual se obtiene un texto con los caracteres de las hojas de respuestas como se observa en la Fig. 3, este texto se compara con las respuestas ingresadas por el docente en el sistema, el algoritmo verifica cuales son las respuestas correctas y asigna la nota que obtuvo en estudiante, en caso de que el algoritmo no pueda procesar una pregunta se presenta como inconsistencia para que el docente revise cual es el inconveniente con esa pregunta.

```

1 @BCD
2 ABC@
3 A.CD
4 A. CD
5 .BCD
6 AB. D
7 ABC.
8 A. CD
9 AB@D
10 ABC.
    
```

Fig. 3 Texto que se obtiene de aplicar OCR a la hoja de repuestas

Antes de obtener los resultados finales se realizaron pruebas con la librería Tesseract, con el propósito de conocer el mejor comportamiento de la librería para reconocer los caracteres de los esquemas de respuestas, se utilizó cinco tipos de esquemas que fueron llenados a mano con esferográfico negro como se observa en la Tabla 1.

Tabla 1. Tipos de Esquemas Analizados

Esquema	Descripción	Captura de la imagen de los esquemas															
Esquema 1	Rellenando la respuesta correcta	<table border="1"> <tr> <td>1</td> <td>○</td> <td>●</td> <td>○</td> <td>○</td> </tr> <tr> <td>2</td> <td>○</td> <td>○</td> <td>●</td> <td>○</td> </tr> <tr> <td>3</td> <td>○</td> <td>●</td> <td>○</td> <td>○</td> </tr> </table>	1	○	●	○	○	2	○	○	●	○	3	○	●	○	○
1	○	●	○	○													
2	○	○	●	○													
3	○	●	○	○													

Esquema 2	Encerrando la respuesta correcta	
Esquema 3	Tachando la respuesta correcta	
Esquema 4	Rellenado la respuesta correcta, representada por números	
Esquema 5	Rellenado la respuesta correcta, representada por letras	

4.2 Resultados y Discusión

Al analizar los distintos esquemas para las hojas de respuestas mediante el método que descarta caracteres, utilizando la librería Tesseract y un filtro gráfico de escala de grises se obtuvo los siguientes resultados:

Tabla 2. Resultados del Esquema 1

Nombre	Pre-gunta	Foto Normal	Escala de Grises
Esquema 1: Rellenando la respuesta correcta	1	✓	x
	2	x	✓
	3	✓	✓
	4	x	✓
	5	x	x
	6	✓	✓
	7	x	✓
	8	✓	✓
	9	✓	✓
	10	x	x
Efectividad:		50%	70%

Tabla 3. Resultados del Esquema 2

Nombre	Pre-gunta	Foto Normal	Escala de Grises
	1	✓	✓

Esquema 2: Encerrando la respuesta co- rrecta.	2	✓	✓
	3	✓	✓
	4	✓	✓
	5	✓	x
	6	✓	✓
	7	✓	✓
	8	✓	✓
	9	✓	✓
	10	✓	✓
	Efectividad:		100%

Tabla 4. Resultados del Esquema 3

Nombre	Pre- gunta	Foto Normal	Escala de Grises
Esquema 3: Tachando la res- puesta	1	✓	✓
	2	x	✓
	3	✓	✓
	4	x	x
	5	✓	✓
	6	✓	✓
	7	x	✓
	8	✓	✓
	9	✓	✓
	10	✓	✓
Efectividad:		70%	90%

Tabla 5. Resultados del Esquema 4

Nombre	Pre- gunta	Foto Normal	Escala de Grises
Esquema 4: Rellenado la res- puesta correcta con números	1	✓	x
	2	✓	x
	3	x	x
	4	x	x
	5	x	x
	6	✓	x
	7	x	x
	8	x	x
	9	x	x
	10	x	x
Efectividad:		30%	0%

Tabla 6. Resultados del Esquema 5

Nombre	Pre- gunta	Foto Normal	Escala de Grises
Esquema 5: Re- llenando la res- puesta correcta con letras.	1	✓	✓
	2	x	✓
	3	✓	✓
	4	x	x
	5	x	✓
	6	x	✓
	7	x	✓
	8	x	✓
	9	x	✓
	10	x	✓
Efectividad:		20%	90%

Resultado final de los Esquemas:

Luego de obtener la efectividad de los esquemas como se observa en la Tabla 6, el esquema que más se adapta a nuestro sistema, es el Esquema 2, con una efectividad del 100% en una fotografía normal y sin aplicar escala de grises como filtro de imagen, este esquema se utilizara para realizar las pruebas funcionales del sistema.

Tabla 7. Resultado Final de los Esquemas

Esquemas	Efectividad	
	Foto normal	Escala de Grises
Esquema 1	50%	70%
Esquema 2	100%	90%
Esquema 3	70%	90%
Esquema 4	30%	0%
Esquema 5	20%	90%

4.2 Resultados de Pruebas Funcionales

Las pruebas funcionales se realizaron en una Unidad Educativa con una población de 100 estudiantes, evidenciándose que se debe tomar en cuenta aspectos técnicos que se escapan de la capacidad del sistema informático, por ejemplo, si el llenado de las hojas de respuestas se realiza con esferográfico azul como se muestra en la Fig 9. El sistema no reconocerá ningún carácter pues Tesseract no está diseñado para detectar una imagen con un color distinto al negro, una posible solución es aplicar un filtro de escala de grises para transformar el color azul a negro, sin embargo, al aplicar el filtro a una imagen llenada con esferográfico negro se pierde efectividad al reconocer los caracteres perjudicando notablemente el funcionamiento del sistema.

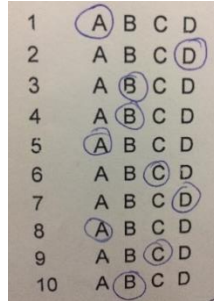


Fig. 4. Hoja de respuestas de la evaluación aplicada llenado con esferográfico color azul

Otro de los problemas que se presentaron al realizar las pruebas, son la forma del círculo en el llenado de las hojas de respuestas como se puede ver en la Fig 10. A pesar de que están llenas con esferográficos negro, el sistema no detecta el carácter correcto cuando el círculo no se encuentra bien definido por lo cual se ve afectado la efectividad del sistema, de igual manera si la resolución de la cámara es muy baja el sistema no reconoce todos los caracteres, la captura de la imagen también es importante pues debe ser a una distancia que cubra todas las respuestas y evitar el movimiento al momento de capturar la foto, todos estos aspectos intervienen negativamente en correcto funcionamiento del sistema.

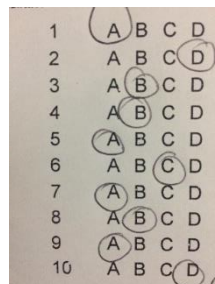


Fig. 5. Hoja de respuestas con círculos que no se encuentran bien definido

El resultado de la ejecución de una prueba se puede ver en la Tabla 7. En donde se tiene una hoja de respuestas llenada por una alumna y al ejecutar el sistema reconoce los caracteres que se pueden visualizar en la misma tabla, el algoritmo que se implementó para la calificación, se encarga de recuperar las posiciones en donde no se encuentra un carácter que pertenezca a las opciones como puede ser A, B o C, depende de la cantidad de opciones de respuestas que el usuario escogió, una vez que el sistema realiza este proceso lo almacena en un arreglo para compararlo con las respuestas que el docente ingreso de esa evaluación en el sistema. En caso de que el sistema detecte un error en una pregunta, se presenta como inconsistencia y el docente ingresara manualmente la respuesta para ejecutar la calificación de la evaluación.

Tabla 8. Resultado obtenido del sistema TesstApp al realizar el proceso de reconocimiento óptico de caracteres

Foto Capturada	Caracteres de la foto	Resultado de la aplicación OCR	Caracteres de la aplicación	Efectividad
1	A	1 A B C D	A	100%
2	D	2 ABC.	D	100%
3	B	3 A.CD	B	100%
4	C	4 AB@D	C	100%
5	A	5 'BCD	A	100%
6	C	6 AB@D	C	100%
7	A	7 .BCD	A	100%
8	B	8 A.CD	B	100%
9	C	9 AB.D	C	100%
10	D	10 ABC.	D	100%
Total:				100%

Resultados de la Evaluaciones

Se realizaron cuatro evaluaciones con un grupo diferente de alumnos cada una de ellas, cada evaluación consta de 4 opciones A, B, C o D y un total de 10 preguntas, los resultados muestran la efectividad del sistema al realizar el OCR. Los resultados finales de todas las evaluaciones se muestran en la Tabla 9.

Tabla 9. Resultados Finales de las Evaluaciones

Test	Alumnos	Hojas de respuestas color negro	Hojas de respuestas otro color	Efectividad
Evaluación 1	22	18	4	62%
Evaluación 2	29	21	8	67%
Evaluación 3	31	25	6	78%
Evaluación 4	18	22	0	96%

Al obtener los resultados se pudo observar que la efectividad del sistema varia entre el 70% y 100% cuando las hojas de respuestas son llenadas con esferográfico negro, sin embargo, la efectividad total de las evaluaciones se ve afectada por las hojas de respuestas que son llenadas de otro color de esferográfico, excluyendo la Evaluación 4 la misma que presenta la efectividad más alta con un 96%, debido al correcto seguimiento de las instrucciones por parte de los alumnos al momento de llenar las hojas de respuestas, esto permite poner a prueba al sistema con las condiciones correctas, el resultado final es de un 96% de efectividad.

5 Conclusiones

Después del respectivo análisis de herramientas, librerías y algoritmos, la librería Tesseract fue la herramienta escogida para el desarrollo del proyecto, Tesseract permitió obtener de manera correcta los caracteres de las hojas de respuestas siempre y cuando cumplieran con las condiciones ideales. La librería Tesseract no reconoce los caracteres cuando son llenados con cualquier tipo de esferográfico que no sea negro, esto se debe a que la librería no reconoce caracteres de color si no solo en escala de grises. Cuando la hoja de respuestas es llenada con esferográfico negro el sistema muestra una efectividad alta al reconocer caracteres. El sistema “TestApp” se desarrolló con herramientas libres y bajo la licencia GPL de software libre, esto conlleva a reducir los costos del sistema sin perder la calidad del mismo.

References

1. Isaza, C., Vargas, J., Gaviria, C., Hernandez, L.: Automatic OCR system for colombian DNIs. STSIVA 2012 - 17th Symp. Image, Signal Process. Artif. Vis. 295–300 (2012).
2. Arrighi, T., Rojas, J.E., Soto, J.C., Madrigal, C. a, Londono, J. a: Recognition and classification of numerical labels using digital image processing techniques. 252–260 (2012).
3. Tejas, B., Omkar, D., Rutuja, D., Prajakta, K., Bhakti, P.: Number plate recognition and document verification using feature extraction OCR algorithm. In: 2017 International Conference on Intelligent Computing and Control Systems (ICICCS). pp. 1317–1320. IEEE (2017).
4. Vuong, B.Q., Do, H.N.: Design and implementation of multilanguage name card reader on Android platform. Int. Conf. Adv. Technol. Commun. 2015–Febru, 528–531 (2015).
5. Ministerio de Educación: REGLAMENTO GENERAL A LA LEY ORGÁNICA DE EDUCACIÓN INTERCULTURAL. , Quito (2015).
6. Cabrera, J.M.M., Sánchez, A.B., Valles, G.Z., Guerrero, S.E., Correa, U.R.G.: SISTEMA DE RECONOCIMIENTO DE DÍGITOS MANUSCRITOS UTILIZANDO REDES NEURONALES. Pist. Educ. 38, (2018).
7. David, O., Santos, B.: Reconocimiento óptico de números escritos a mano usando funciones de base radial y sistema memético diferencial | Piragauta Gómez | Revista vínculos. 68–78 (2014).
8. Palekar, R.R., Parab, S.U., Parikh, D.P., Kamble, V.N.: Real time license plate detection using openCV and tesseract. 2017 Int. Conf. Commun. Signal Process. 40055, 2111–2115 (2017).
9. Ramappa, V., Bein, D.: MusiqGlobe.fm using MEAN stack. 2018 IEEE 8th Annu. Comput. Commun. Work. Conf. CCWC 2018. 2018–Janua, 661–664 (2018).
10. Lanka, S.: A MEAN Stack Based Remote Monitoring System for Electrical Distribution Feeders in Sri Lanka. 3033–3038 (2017).
11. Guo-Hong, S.: Application Development Research Based on Android Platform. In: 2014 7th International Conference on Intelligent Computation Technology and Automation. pp. 579–582. IEEE (2014).