**Documento Técnico: Detección de COVID usando CNN (red neuronal convolucional) mediante Radiografías de Tórax vía Telegram**

**Entidades Interesadas:**

MINTIC, Talento Tech

Universidad de Antioquia

Universidad de caldas

Fecha: 30 de abril de 2025

Versión: 1.0

Autor: Dorian A García Giraldo

Contenido

[**1. Objetivo 3**](#_dz4gdi4btlge)

[**2. Alcance 3**](#_2iec3ffdx9f6)

[**3. Resultado esperado 4**](#_at89hbc6s1y5)

[**4. Cronograma 5**](#_usv0q3hmiokl)

[**5. Presupuesto 5**](#_1uk5hx1wovf0)

[5.1. Licencias y tecnologías 5](#_ulf2w9rs1gj8)

[5.2. Equipo de trabajo 6](#_a1tc783libbh)

[5.3. Presupuesto final 6](#_wl3zkofls5h3)

[**6. Gestión de Riesgos 6**](#_w0pov7rfnkmg)

[**7. Gestión de comunicaciones 7**](#_aiuvtux14era)

[7.1. Sesiones de trabajo 7](#_pcvhyq5euy3c)

[7.2. Canales de comunicación 7](#_fthyofwdcfp3)

[**8. Gestión del equipo 7**](#_wbft6qomqcje)

[**9. Seguimiento del progreso 8**](#_cpn9f0lqo8ej)

[**10. Gestión de cambios 8**](#_hvmlygeukl4i)

[**11. Gestión de Calidad 8**](#_igirlf81eteu)

[**12. Entrega de Resultados 9**](#_8kf34ea6ndab)

[**13. Evaluación y Lecciones Aprendidas 9**](#_uihk7p1zx67s)

[**14. Cierre Administrativo 9**](#_g6doh9k94k5n)

Definición del Proyecto

1. **Objetivo**

Desarrollar e implementar un sistema automatizado para la detección de posibles signos de COVID en radiografías de tórax cargadas por usuarios a través de la plataforma de mensajería Telegram. El sistema utilizará una red neuronal convolucional (CNN) previamente entrenada para analizar las imágenes y proporcionar una predicción sobre la presencia o ausencia de indicadores de COVID, junto con un nivel de confianza asociado.

1. **Alcance**

El proyecto abarca las siguientes fases y funcionalidades:

* **Desarrollo del Bot de Telegram:** Creación de un bot de Telegram que permita a los usuarios cargar imágenes de radiografías de tórax.
* **Integración con Modelo de Detección:** Implementación de la lógica para recibir la imagen del usuario, preprocesarla y enviarla a un modelo de red neuronal convolucional (CNN) previamente entrenado para la detección de COVID.
* **Preprocesamiento de Imágenes:** Desarrollo de funciones para el preprocesamiento de las imágenes recibidas (redimensionado, normalización, conversión a escala de grises, etc.) para asegurar la compatibilidad con el modelo de detección.
* **Análisis con Red Neuronal:** Utilización del modelo de CNN para analizar la imagen preprocesada y obtener una predicción (presencia/ausencia de COVID) y un nivel de confianza.
* **Generación de Respuesta:** Creación de mensajes claros y concisos para informar al usuario sobre el resultado del análisis, incluyendo la predicción y el nivel de confianza.
* **Implementación y Despliegue:** Despliegue del bot de Telegram y la infraestructura necesaria para el funcionamiento del sistema.
* **Documentación Técnica:** Elaboración de la documentación técnica detallada del sistema, incluyendo la arquitectura, las APIs utilizadas, el proceso de implementación y las consideraciones de despliegue.

**El proyecto no incluye:**

* Entrenamiento del modelo de red neuronal convolucional. Se asumirá que un modelo previamente entrenado con un rendimiento adecuado está disponible.
* Validación clínica de los resultados del sistema. El sistema se presenta como una herramienta de apoyo y no como un diagnóstico médico definitivo.
* Integración con sistemas de información hospitalarios o registros médicos electrónicos.
* Desarrollo de funcionalidades avanzadas de procesamiento de imágenes o análisis.

**3. Resultado Esperado**

Se espera obtener un bot de Telegram funcional que permita a los usuarios cargar radiografías de tórax y recibir una respuesta automatizada indicando la posible presencia o ausencia de signos de COVID, junto con un nivel de confianza. Este sistema proporcionará una herramienta inicial de detección que podría ser útil para la priorización de casos o como una herramienta educativa.

**4. Cronograma**

El cronograma detallado del proyecto se definirá en una etapa posterior, pero se espera que incluya las siguientes fases principales:

* **Fase 1: Planificación y Diseño (Semana 1-2)**
* **Fase 2: Desarrollo del Bot de Telegram e Integración con Modelo (Semana 3-6)**
* **Fase 3: Desarrollo del Preprocesamiento de Imágenes (Semana 4-5)**
* **Fase 4: Desarrollo de la Lógica de Predicción y Respuesta (Semana 6-7)**
* **Fase 5: Pruebas Internas (Semana 8)**
* **Fase 6: Despliegue (Semana 9)**
* **Fase 7: Documentación Final (Semana 10)**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **Semanas** | | | | | | |
|  | Actividades | **S1** | **S2** | **S3** | **S4** | **S5** | **S6** | **S7** |
|  | 20/03 | 27/03 | 4/05 | 11/05 | 18/05 | 25/05 | 31/5 |
| **Misión 1** | Crear equipo de trabajo |  |  |  |  |  |  |  |
| Diseña Clases Principales |  |  |  |  |  |  |  |
|  | Crear Diagrama UML |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| **Misión 2** | Desarrollar Gestión de Usuarios |  |  |  |  |  |  |  |
| Desarrollar Interfaces de visualización |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| **Misión 3** | Diseñar e implementar pruebas |  |  |  |  |  |  |  |
| Socializar primera versión |  |  |  |  |  |  |  |
| Ajustes finales |  |  |  |  |  |  |  |
| Lecciones aprendidas |  |  |  |  |  |  |  |

**5. Presupuesto**

El presupuesto detallado se definirá en una etapa posterior, pero se consideran las siguientes categorías principales:

* **5.1. Licencias y Tecnologías:**
  + Costo de licencias de software (si aplica).
  + Costo de servicios en la nube para el despliegue (si aplica).
  + Costo de APIs de Telegram (si aplica, aunque la API básica es gratuita).
* **5.2. Equipo de Trabajo:**
  + Honorarios de desarrolladores.
  + Honorarios de testers (si aplica).
  + Costos de gestión de proyecto.
* **5.3. Presupuesto Final:** Sumatoria de los costos anteriores, incluyendo un margen para imprevistos.

**6. Gestión de Riesgos**

Se identifican los siguientes riesgos potenciales:

* **Disponibilidad y rendimiento del modelo de detección:** Si el modelo preexistente no funciona como se espera o tiene un rendimiento insuficiente, esto podría afectar la precisión del sistema.
* **Problemas de integración con la API de Telegram:** Cambios en la API de Telegram o dificultades en la implementación podrían retrasar el desarrollo del bot.
* **Errores en el preprocesamiento de imágenes:** Un preprocesamiento incorrecto podría llevar a resultados erróneos en la detección.
* **Problemas de despliegue:** Dificultades en la configuración del entorno de despliegue podrían retrasar la puesta en marcha del sistema.
* **Cambios en los requisitos:** Solicitudes de nuevas funcionalidades o cambios en los requisitos iniciales podrían afectar el cronograma y el presupuesto.

Para mitigar estos riesgos, se implementarán las siguientes acciones:

* Selección y prueba exhaustiva del modelo de detección antes de la integración.
* Seguimiento constante de la documentación y posibles cambios en la API de Telegram.
* Desarrollo y prueba rigurosa de las funciones de preprocesamiento de imágenes.
* Planificación cuidadosa del proceso de despliegue y pruebas en un entorno similar al de producción.
* Gestión proactiva de los requisitos y comunicación constante con los stakeholders.

**7. Gestión de Comunicaciones**

La comunicación efectiva entre los miembros del equipo y los stakeholders es crucial para el éxito del proyecto. Se utilizarán los siguientes mecanismos:

* **7.1. Sesiones de Trabajo:** Reuniones periódicas del equipo para discutir el progreso, los desafíos y las próximas tareas. La frecuencia y el formato se definirán en la fase de planificación.
* **7.2. Canales de Comunicación:**
  + **Correo electrónico:** Para comunicaciones formales y documentación.
  + **Plataforma de mensajería instantánea:** Para comunicaciones rápidas y coordinación del equipo.
  + **Herramientas de gestión de proyectos:** Para el seguimiento de tareas, la gestión de issues y la documentación del progreso.

**8. Gestión del Equipo**

El equipo de trabajo estará compuesto por:

* **[Roles y responsabilidades de los miembros del equipo, por ejemplo:]**
  + Líder del Proyecto: Responsable de la planificación, ejecución y seguimiento del proyecto.
  + Desarrollador Backend: Responsable del desarrollo del bot de Telegram, la integración con el modelo y la lógica de la aplicación.
  + Científico de Datos/Ingeniero de Machine Learning: Responsable de la comprensión e integración del modelo de detección.
  + Tester (Opcional): Responsable de la planificación y ejecución de las pruebas del sistema.

**9. Seguimiento del Progreso**

El progreso del proyecto se monitorizará utilizando las siguientes métricas e hitos:

* Finalización de las tareas definidas en el plan de trabajo.
* Cumplimiento del cronograma establecido.
* Número de funcionalidades implementadas y probadas.
* Identificación y resolución de issues y riesgos.
* Entrega de los entregables definidos.

Se realizarán informes de progreso periódicos para comunicar el estado del proyecto a los stakeholders.

**10. Gestión de Cambios**

Cualquier solicitud de cambio en el alcance, los requisitos o el cronograma del proyecto se gestionará de acuerdo con un proceso formal de gestión de cambios. Esto incluirá la evaluación del impacto del cambio, la aprobación por parte de los stakeholders relevantes y la actualización de la documentación del proyecto.

**11. Gestión de Calidad**

Se implementarán las siguientes medidas para asegurar la calidad del sistema:

* **Revisiones de código:** Para identificar posibles errores y asegurar el cumplimiento de los estándares de codificación.
* **Pruebas unitarias:** Para verificar la correcta funcionalidad de los componentes individuales del sistema.
* **Pruebas de integración:** Para asegurar la correcta interacción entre los diferentes módulos del sistema (bot de Telegram, preprocesamiento, modelo de detección).
* **Pruebas de usuario (si aplica):** Para obtener retroalimentación de los usuarios y asegurar que el sistema cumple con sus necesidades.

**12. Entrega de Resultados**

Los principales entregables del proyecto son:

* Código fuente del bot de Telegram y la lógica de integración.
* Documentación técnica del sistema (este documento y otros documentos de diseño e implementación).
* Infraestructura desplegada (si aplica).
* Informe final del proyecto.

**13. Evaluación y Lecciones Aprendidas**

Una vez finalizado el proyecto, se realizará una evaluación para identificar los éxitos, los desafíos y las lecciones aprendidas durante el desarrollo. Esta evaluación se documentará y se utilizará para mejorar futuros proyectos.

**14. Cierre Administrativo**

La fase de cierre administrativo incluirá la formalización de la finalización del proyecto, la entrega final de la documentación, la obtención de la aprobación final de los stakeholders y la realización de cualquier actividad administrativa necesaria para cerrar el proyecto.

Se entregará una carpeta ZIP con todo el proyecto.

**Anexo: Pasos para el Procesamiento de la Imagen**

El siguiente diagrama describe los pasos principales para el procesamiento de la imagen cargada desde Telegram hasta su análisis por el modelo de detección:

Fragmento de código

graph TD

A[Usuario carga radiografía en Telegram] --> B(Bot de Telegram recibe la imagen);

B --> C{¿Imagen válida?};

C -- Sí --> D[Descargar archivo de la imagen];

D --> E[Leer la imagen (PIL/OpenCV)];

E --> F[Convertir a escala de grises];

F --> G[Redimensionar a dimensiones del modelo (IMG\_WIDTH x IMG\_HEIGHT)];

G --> H[Normalizar los valores de píxeles (0-1)];

H --> I[Expandir dimensiones para el modelo (batch\_size, height, width, channels)];

I --> J[Enviar imagen preprocesada al modelo de CNN];

J --> K[Modelo realiza la predicción];

K --> L{¿Confianza > Umbral?};

L -- Sí --> M[Generar respuesta: Posible COVID];

L -- No --> N[Generar respuesta: No se detectaron signos claros];

M --> O[Enviar respuesta al usuario por Telegram];

N --> O;

C -- No --> P[Enviar mensaje de error al usuario];

P --> O;

**Descripción Detallada de los Pasos:**

1. **Usuario carga radiografía en Telegram:** El usuario utiliza la interfaz de Telegram para enviar una imagen al bot.
2. **Bot de Telegram recibe la imagen:** El bot, a través de la API de Telegram, recibe el archivo de la imagen.
3. **¿Imagen válida?:** Se realiza una verificación básica para asegurar que el archivo recibido es una radiografía.
4. **Descargar archivo de la imagen:** El bot descarga el archivo de la imagen desde los servidores de Telegram.
5. **Leer la imagen (PIL/OpenCV):** Se utiliza una librería de procesamiento de imágenes como PIL (Pillow) o OpenCV para cargar la imagen en la memoria.
6. **Convertir a escala de grises:** Si el modelo fue entrenado con imágenes en escala de grises, la imagen se convierte a un solo canal de intensidad.
7. **Redimensionar a dimensiones del modelo (IMG\_WIDTH x IMG\_HEIGHT):** La imagen se redimensiona a las dimensiones específicas con las que fue entrenado el modelo de CNN. Esto es crucial para la compatibilidad.
8. **Normalizar los valores de píxeles (0-1):** Los valores de los píxeles (generalmente en el rango 0-255) se normalizan al rango 0-1 dividiéndolos por 255. Esto ayuda a la convergencia y el rendimiento del modelo.
9. **Expandir dimensiones para el modelo (batch\_size, height, width, channels):** Se añaden dimensiones adicionales a la imagen para que coincida con la entrada esperada por el modelo de CNN (por ejemplo, añadiendo una dimensión de "batch" aunque se procese una sola imagen, y una dimensión de "canal" aunque sea escala de grises).
10. **Enviar imagen preprocesada al modelo de CNN:** La imagen preprocesada se alimenta como entrada a la función de predicción del modelo de CNN.
11. **Modelo realiza la predicción:** El modelo analiza la imagen y genera una salida, que típicamente incluye una probabilidad o un valor de confianza para cada clase (en este caso, presencia o ausencia de COVID).
12. **¿Confianza > Umbral?:** La confianza de la predicción para la clase "COVID" se compara con un umbral predefinido (por ejemplo, 0.6).
13. **Generar respuesta: Posible COVID:** Si la confianza supera el umbral, se genera un mensaje indicando una posible detección de COVID junto con el nivel de confianza.
14. **Generar respuesta: No se detectaron signos claros:** Si la confianza no supera el umbral, se genera un mensaje indicando que no se detectaron signos claros de COVID, posiblemente incluyendo la confianza para la clase negativa.
15. **Enviar respuesta al usuario por Telegram:** El bot envía el mensaje generado al usuario a través de la API de Telegram.
16. **Enviar mensaje de error al usuario:** Si la imagen no es válida en el paso 3, se envía un mensaje de error al usuario pidiéndole que envíe una imagen válida.