PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

Escuela de Posgrado

Modelos de Detección de Emociones en Texto y Rostros para Agentes Conversacionales Multimodales

Tesis para obtener el grado académico de Magíster en Informática con Mención en Ciencias de la Computación que presenta:

José Guillermo Balbuena Galván

Asesor:

Dr. Cesar Armando Beltran Castañon

Lima, 2022

RESUMEN

El presente trabajo de investigación aborda la implementación, análisis y selección de distintos modelos de redes neuronales recurrentes (RNN) y convolucionales (CNN) para la detección de emociones en texto y rostros; los cuales pueden ser utilizados como módulos adicionales en agentes conversacionales de tiempo real como son chatbots o robots sociales. Los módulos de detección permiten a los agentes conversacionales poder entender cómo se sienten las personas durante la interacción con ellas; conociendo estos estados los agentes conversacionales pueden responder empáticamente.

En primer lugar, se revisará la literatura sobre como los agentes conversacionales buscan ser más empáticos, así como los métodos de detección de emociones mediante distintos canales como texto y rostros. Luego, se procede a recolectar y pre-procesar bases de datos públicas para el entrenamiento de los algoritmos seleccionados en base a la literatura.

Finalmente, métricas tanto para la evaluación del rendimiento de predicción multiclase (*Accuracy, Precision, Recall y F1*), como la velocidad de procesamiento (ej. *Frames-per-second*) son seleccionadas y analizadas para determinar cuáles son los mejores algoritmos para implementar una aplicación de tiempo real.

ÍNDICE DE CONTENIDO

Pág.

ÍNDICE DE TABLAS

Pág.

ÍNDICE DE FIGURAS

Pág.

CAPÍTULO 1

GENERALIDADES

1.1 Introducción

Los *Agentes Conversacionales* (AC) y *Chatbots* son una de las modalidades por las cuales las personas interactuamos con las computadoras, y según *J. Wirtz et al.* (2018) pueden ser clasificados como un tipo de robot de servicio virtual que interactúa con las personas mediante acciones intangibles. Asimismo, *J. Wirtz* menciona que los robots pueden completar un gran volumen de trabajo cuando este es homogéneo, volviendo esta tecnología atractiva para el comercio electrónico.

No obstante, su uso no se limita a sectores donde grandes volúmenes de trabajo son procesados a diario como: "*Asistencia Técnica*", "*Servicio al Cliente*", "*Ayuda en ventas*", "*Mesa de Ayuda*" (Lester, Branting, & Mott, 2004). Esta tecnología también ha sido aplicada en el sector educación, médico y psicológico; siendo utilizada como una herramienta para tratamiento psicológico (Fitzpatrick, Darcy, & Vierhile, 2017; Jha, Khant, Kotadiya, Gamdha, & Kansagra, 2019; Kataria, Rode, Jain, Dwivedi, & Bhingarkar, 2018; Oh, Lee, Ko, & Choi, 2017; Sharma, Puri, & Rawat, 2018) o como un asistente personalizado para la enseñanza de cursos en línea (Holotescu, 2016).

El incremento en el uso de este tipo de tecnologías y los avances en las áreas como la robótica, realidad virtual, ciencias de la computación e inteligencia artificial, han incrementado el interés en la investigación relacionada a la computación afectiva. La *Computación Afectiva* es una rama multidisciplinaria que involucra a distintas

especialidades entre ellas psicología y ciencias de la computación; el objetivo es brindar a las computadoras la capacidad que poseen los humanos para reconocer, procesar y expresar emociones (Strauss et al., 2005); ayudando a mejorar la Interacción Humano-Computador (*HCI*, por sus siglas en inglés); volviendo el dialogo más real.

1.2 Problemática

El uso de teléfonos inteligentes e Internet ha incrementado en los últimos años, sobre todo en los países en vía de desarrollo como Perú, según el estudio presentado por Poushter (2016), aproximadamente en el 2015, el 67% de la población tiene acceso a internet y el 43% posee un teléfono inteligente. Asimismo, la popularidad de las redes sociales ha aumentado llegando a un promedio de 76% a nivel mundial entre los usuarios de internet.

Este incremento en el acceso a internet ha llevado al incremento de usos de servicios en línea, y por consiguiente a la interacción con agentes conversacionales y chatbots. Por ejemplo, las empresas han comenzado a desarrollar agentes conversacionales para funciones como servicio al cliente en redes sociales (Xu, Liu, Guo, Sinha, & Akkiraju, 2017) o para el comercio electrónico (Van Eeuwen, 2017), disminuyendo cada vez más el uso de las tradicionales vías telefónicas (Hyken, 2017).

Existen distintas razones para la inclusión de emociones a los agentes conversacionales (Becker, Kopp, & Wachsmuth, 2007) entre ellas el desarrollo de un *agente-creíble* y la *experimentación-teórica*, pero el aspecto de la interacción social ha ganado mayor fuerza en los últimos años, puesto que las personas han comenzado a ver a las computadora como actores sociales.

P. B. Brandtzaeg et al (2017) menciona que las interacciones humano-computadora (chatbot) tienden a durar más en comparación a una conversación entre dos personas desconocidas, asimismo las personas usan con las computadoras frases cortas y evitan lenguaje complejo. A esto se suma el hallazgo realizado por Xu et al (2017), que el 40% de personas en Twitter realiza solicitudes emocionales a los servicios al cliente ofrecidos, revelando un nuevo paradigma para la interacción. En este sentido la necesidad de investigaciones relacionadas al desarrollo algoritmos de detección para agentes *conversacionales emocionales* es necesaria y esencial, debido a la cotidianidad con la que recurrimos al uso de estas tecnologías.

1.3 Objetivo

Seleccionar algoritmos para detección de emociones en texto y rostros que sirvan como módulos para un agente conversacional. Los modelos deben ser seleccionados en base a un balance entre métricas que midan la efectividad de los modelos y su capacidad de ser usados en tiempo real, para ser implementados en plataformas robóticas.

1.3.1 Objetivos Específicos

- Recolectar los conjuntos de datos públicos necesarios para el desarrollo de los algoritmos (imágenes y textos), y pre-procesarlos para el entrenamiento de los modelos.
- Seleccionar un algoritmo de clasificación de emociones en texto utilizando *Redes Neuronales Recurrentes*.
- Seleccionar un algoritmo basado en una *Red Neuronal Convolucional* para la clasificación de emociones utilizando imágenes digitales de las expresiones faciales.
- Evaluar las métricas de evaluación para cada algoritmo y comparar resultados para las distintas formas de detección de emociones.

1.4 Metodología

El presente trabajo de investigación se desarrollará en cuatro (04) etapas, que permitirán el correcto desarrollo de los objetivos mencionados.

a. **Revisión del estado del arte:** en esta etapa se realizará una revisión de todas las investigaciones relacionadas a la generación de emociones en agentes conversacionales de dominio abierto y a métodos para la detección de emociones mediante texto y rostros.

b. **Recolección y procesamiento de datos:** en esta etapa se recolectarán base de datos (textos e imágenes) de dominios públicos que permitan el entrenamiento de los algoritmos a desarrollar.

c. **Desarrollo de los módulos de detección para el agente conversacional emocional:** en esta etapa desarrolla la estructura del agente la cual consta de dos pasos principales: 1) La detección de las emociones mediante el uso de texto; y

2) Detección de emociones utilizando los rostros en imágenes utilizando distintos algoritmos.

d. **Selección de métricas y evaluación de algoritmos:** en esta etapa se seleccionarán las métricas adecuadas para los algoritmos seleccionados. Asimismo, se evaluara su desempeño en tiempo real evaluando cuantas inferencias por segundo pueden realizar y que factores influyen en estos (longitud de las oraciones y tamaño de las imágenes)

1.5 Alcance

El alcance del presente trabajo de investigación es el diseño, construcción y evaluación de modelos de detección de emociones en texto y rostros que sirvan como módulos para agentes conversacionales con la característica de ser simples y poder ser ejecutados en sistemas con bajo poder computacional. Se realizarán una evaluación de inferencias por segundo para determinar su desempeño en aplicaciones en tiempo real.

CAPÍTULO 2

MARCO TEÓRICO

En este capítulo se presentarán los conceptos básicos relacionados a los agentes conversacionales, emociones y distintas técnicas del aprendizaje profundo para imágenes e información secuencial.

2.1 Agente Conversacional

Los agentes conversacionales (AC) pueden ser definidos como sistemas automáticos con la capacidad de emular a una persona durante el dialogo, son capaces de entender al usuario y responder (Griol, Sanchis, Molina, & Callejas, 2019). A diferencia de los chatbots, los AC deben poder manejar conversaciones variables y no limitarse a responder a preguntas o comandos. Otras definiciones, presentan a los AC como la integración de técnicas lingüísticas para entender el lenguaje natural permitiendo responder oraciones y seguir diálogos (Lester et al., 2004), estas características de los AC han permitido ser utilizado en distintas aplicaciones como: "*Help-desk*", "*Servicio al cliente*", "*Soporte Técnico*", entre otros.

Los agentes conversacionales tienen una estructura básica, ilustrada en la Figura 2.1, que se puede dividir en tres (03) componentes: 1) El *Interpreter*, que se encarga de analizar la oración de que ha sido ingresada por el usuario. 2) El *Dialog Manager*, que es el encargado de decidir qué acción realizar para responder al usuario. 3) El *Response Generator*, que es el encargado de producir una respuesta coherente al usuario usando los canales que tenga disponibles (texto, voz, etc.)

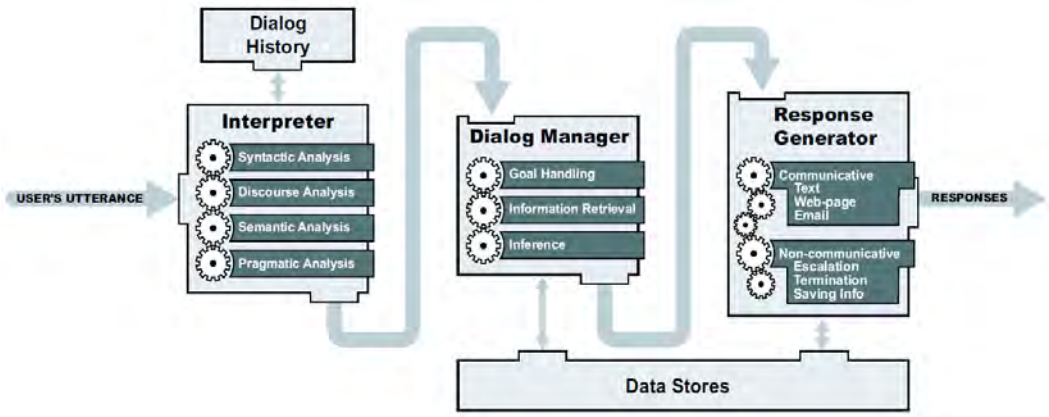


Figura 2.1. Componentes principales de un agente conversacional o chatbot.
Fuente: "Conversational Agents"

2.1.1 Clasificación de agentes conversacionales

Los agentes conversacionales al igual que los chatbots pueden dividirse de distintas maneras de acuerdo a diferentes criterios (Hussain, Sianaki, & Ababneh, 2019) entre ellos: modo de interacción (texto o voz), conocimiento del dominio (dominio especifico o abierto), método de generación de respuesta (basado en reglas o AI) y la aplicación (orientado o no a tareas). A continuación, se describirán algunas de las clasificaciones:

a. Task Oriented / Non Task Oriented:

Los agentes conversacionales *Task-Oriented* tienen como objetivo ayudar al usuario a completar una tarea específica como: realizar reservaciones, programar eventos, ejemplos claros de este tipo son Cortana, Alexa y Siri que utilizan la voz como medio de comunicación. Este tipo de agentes trabajan bien en *dominios específicos*; puesto que no poseen conocimientos distintos a los especificados.

Por otro lado, los *Non Task-Oriented* se utilizan para conversaciones largas sin una estructura de dialogo, como la conversación natural entre humanos y usualmente utilizados en contextos de *Open-Domain* y utilizan entre otros *Generative Rules* para la generación de respuestas

2.1.2 Métodos para generación de respuestas

La cantidad de métodos para la elaboración de agentes conversacionales o chatbots es amplia, de acuerdo a Hussain et al. (2019) pueden ser divididos en tres (03) categorías principales:

- Basado en Reglas o *Ruled-Based*: este tipo de agente conversacional usa un set de reglas pre-definidas (Joshi, 2020) y puede tener distintas complejidades. Estos son desarrollados en base a instrucciones condicionales `if/else` permitiendo un entrenamiento rápido, en la Figura 2.2 se muestra un ejemplo del flujo de trabajo de este tipo de método.

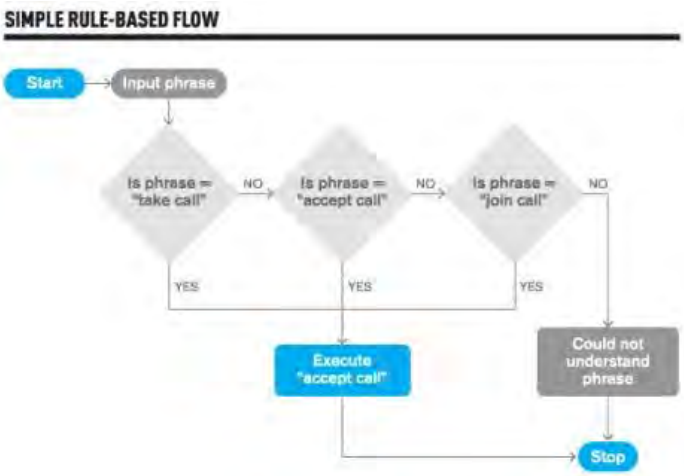


Figura 2.2. Ejemplo de una aproximación *Ruled-Based*.
Fuente: medium.com

- *Basado en Recuperación o Retrieval-Based:* este tipo de agente conversacional el algoritmo aprende a seleccionar su respuesta de una lista pre-definida y que puede ser revisada de acuerdo a la calidad, fluidez y diversidad (Swanson, Yu, Fox, Wohlwend, & Lei, 2019). Varios utilizan una lista de preguntas-respuestas para lograr un largo repositorio (Lommatzsch & Katins, 2019). En la Figura 2.3 se pueden observar un modelo simple de esta aproximación, las técnicas usuales incluyen redes neurales recurrentes (RNN) y sus variantes.

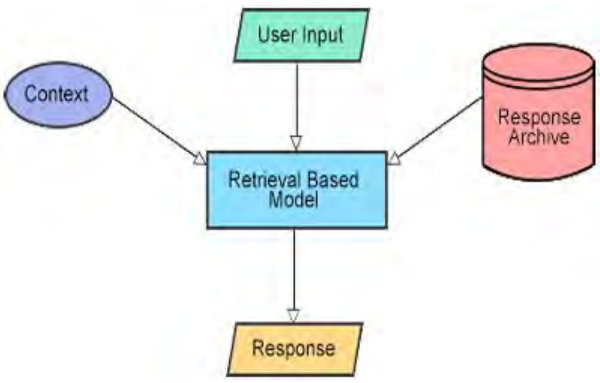


Figura 2.3. Estructura simple del método *Retrieval-Based*.
Fuente: dzone.com

- *Basado en Generación o Generative-Based:* este tipo puede aprender a generar respuestas correctas que no han aparecen en el corpus o repositorio de entrenamiento. Generalmente, usan técnicas de inteligencia artificial (AI) como los modelos neuronales *sequence-to-sequence* (seq2seq), en la Figura 2.4 se observa cómo trabajan estos modelos.

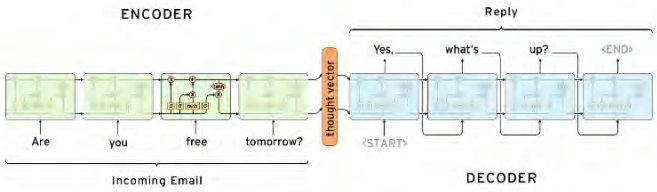


Figura 2.4. Modelo seq2seq usado en la aproximación *Generative-Based*.
Fuente: medium.com

2.2 Emociones

En la literatura no existe un consenso para la definición de que es una emoción (Cabanac, 2016), no obstante varios psicólogos enfatizan en la naturaleza episódica de las emociones (Scherer, 2000); es decir son disparadas por un evento externo o interno. El episodio emotivo debe durar cierto periodo y va decreciendo en intensidad; los problemas en la definición surgen al tratar de consensuar los componentes del episodio.

2.2.1 Modelos de clasificación de emociones

Los modelos utilizados para la clasificación de emociones son por lo general "Categorías Emocionales" y "Dimensiones Emocionales" (Canales, Martinez-Ba, & Rco, 2018). De acuerdo a K. Scherer (Scherer, 2000), los modelos dimensionales se enfocan en los sentimientos subjetivos y estos se diferencian por el grado de similaridad entre dimensiones; por otro lado, las categorías emocionales se enfocan en la expresión motora o patrones de comportamiento. A continuación, se detallan algunos de los modelos más utilizados.

a. Emociones Básicas de Ekman

El modelo presentado por Ekman et al. (1973) presenta seis (06) emociones básicas, las cuales son: Felicidad, Ira, Disgusto, Tristeza, Sorpresa y Miedo. Estas emociones como conclusión del estudio realizado por Ekman et al. (1973), este consistía en que personas de distintas nacionalidades puedan reconocer e identifiquen las distintas emociones basándose en la expresión facial.

b. La rueda de emociones de Plutchik

Plutchik (1982) menciona que para determinar cuáles son las emociones primarias y poder etiquetarlas se tienen que entender en un marco evolutivo, pues se deben aplicar a animales y humanos. Asimismo, sugiere ocho (08) emociones: Miedo, Ira, Alegría, tristeza, Confianza, Disgusto, Sorpresa y Anticipación. Estas emociones se pueden organizar en pares opuestos como se puede ver en la siguiente figura.

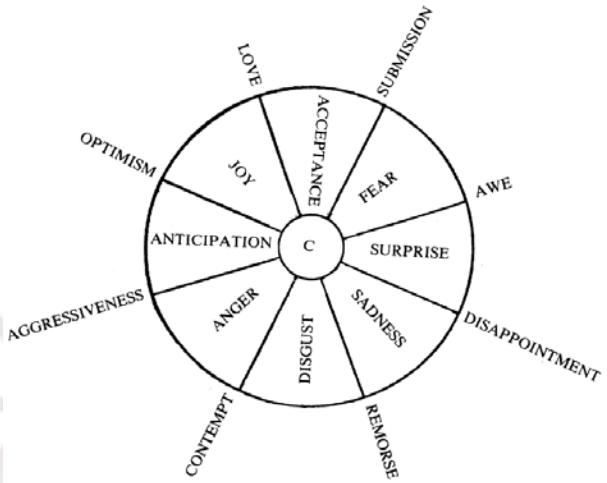


Figura 2.5. Pares adyacentes de las emociones básicas.
Fuente: "A psychoevolutionary theory of emotions"

c. Modelo Circumplex

Estudios han demostrado que las personas no pueden describir sus emociones de manera aislada y de manera discreta, por lo que Posner (2005) detalla una estructura 2D en el cuál las dimensiones son *Arousal* y *Valence*; en este diagrama o estructura cada emoción se describe como una combinación lineal de las dos (02) dimensiones mencionadas. En la Figura 2.6 el eje horizontal representa la dimensión *valence* y el vertical el *aurosal.*

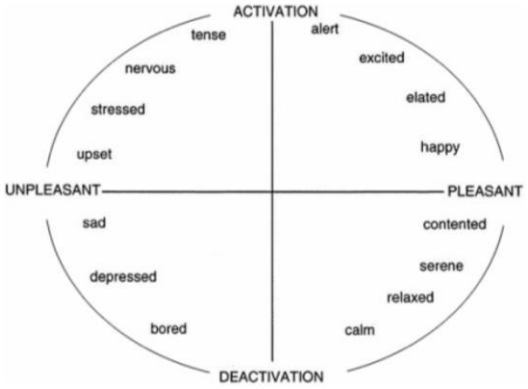


Figura 2.6. Componentes principales de un agente conversacional o chatbot.
Fuente: "The circumplex model of affect: An integrative approach to affective neuroscience, cognitive development, and psychopathology"

2.3 Redes Neuronales Convolucional

Las redes convolucionales, también conocidas como redes neuronales convolucionales (CNN), son un tipo especial de red neuronal para el procesamiento de información del tipo cuadricula (Goodfellow, Bengio, & Courville, 2016) como las imágenes; el nombre deriva del uso de la operación matemática conocida como la "convolución". Las CNN's son propuestas por LeCun et al. (Lecun, Bottou, Bengio, & Ha, 1998) con el objetivo de asegurar un grado de invariancia frente a la distorsión, escalado y traslación que pueden surgir al momento de analizar imágenes. Las CNN's son las arquitecturas con mayor éxito en aplicaciones reales y se manifiesta en las distintas arquitecturas propuestas en la literatura (Bakhshi, Chalup, & Noman, 2020), a continuación se presentarán algunas arquitecturas de CNN.

2.3.1 Arquitectura VGG

Simonyan y Zisserman (2015), proponen una arquitectura llamada VGG en la que buscan incrementar el número de capas de la CNN, llegando hasta 16 y 19; con lo que mejoraron el performance. La arquitectura propuesta usa filtros 3x3 para reducir el número de pesos y el 'stride' de la convolución lo mantienen en 1, en la siguiente figura se pueden ver las distintas arquitectura realizadas por Simonyan et al. (2015).

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Figura 2.7. Arquitecturas CNN's para el VGG
Fuente: "Very Deep Convolutional Networks For Large-Scale Image Recognition"

2.3.2 Arquitectura ResNet

Ante el problema del desvanecimiento de gradiente en las redes profundas al incrementar las capas, He et al. (2016) propone un aprendizaje profundo residual (en inglés *deep residual learning*) para poder incrementar el número de capas en las arquitecturas y mejorar la exactitud. Los bloques residuales utilizados para armar estas arquitecturas se presentan en la Figura 2.8. La utilización de arquitecturas basadas en estos bloques, Residual Net (ResNet), en el CIFAR-10 (Krizhevsky, 2009) demostró que para mayor cantidad de capas se obtenía menor error, por ejemplo una ResNet-20 obtuvo 8.75% y en una ResNet-110 6.43%.

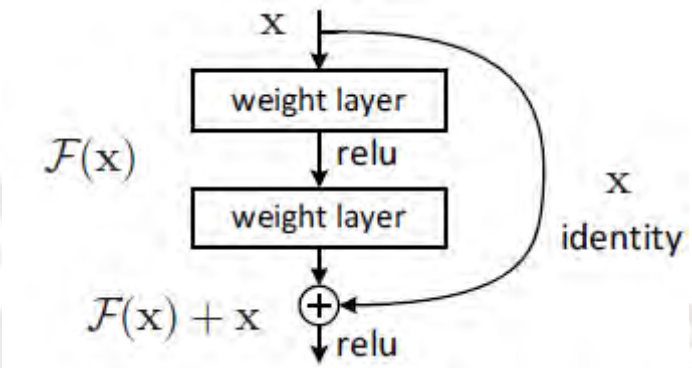


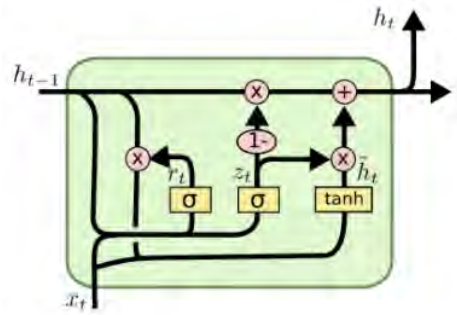
Figura 2.8. Bloque Residual de las ResNet.
Fuente: "Deep Residual Learning for Image Recognition"

2.4 Redes Neuronales Recurrentes

Las redes neuronales recurrentes o RNNs son una familia de redes neuronales especializadas para el procesamiento de información secuencial o secuencia de valores (Goodfellow et al., 2016). Las redes recurrentes han sido utilizadas en distintas áreas como son la representación del lenguaje natural, descripción de imágenes, traductores, generador de diálogos, entre otros (Lipton, Berkowitz, & Elkan, 2015); a continuación, se presentaran algunas de las arquitecturas utilizadas en las RNNs.

2.4.1 Long-Short Term Memory (LSTM)

Ante el problema de las señales tienden a desvanecerse o explotar durante la "propagación hacia atrás a través del tiempo" (en inglés, backpropagation through time - BPTT), Hochreiter y Schmidhuber (1997) proponen las LSTM; esta arquitectura introduce lo que se conoce como celdas de memoria (*memory cells*) y puertas (*gate units*). Las LSTM están diseñadas para poder recordar información por largos periodos en el tiempo (Colah's Blog, n.d.), en la siguiente figura se puede ver las ecuaciones de una LSTM.



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$
$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Figura 2.9. Arquitectura de una LSTM
Fuente: colah.github.io/posts/2015-08-Understanding-LSTMs/

2.4.2 Red Neuronal Recurrente Bidireccional (BRNN)

Schuster and Paliwal (Schuster & Paliwal, 1997) proponen las BRNN para extender la cantidad de información que puede utilizar una RNN, la idea es separar las neuronas en dos (02) partes; unas se encargan de la dirección positiva del tiempo (forward states) y las otras de la negativa (backward states), las salidas de ambos estados no se conectan. En la siguiente figura se puede ver un ejemplo de una BRNN en tres pasos de tiempo.

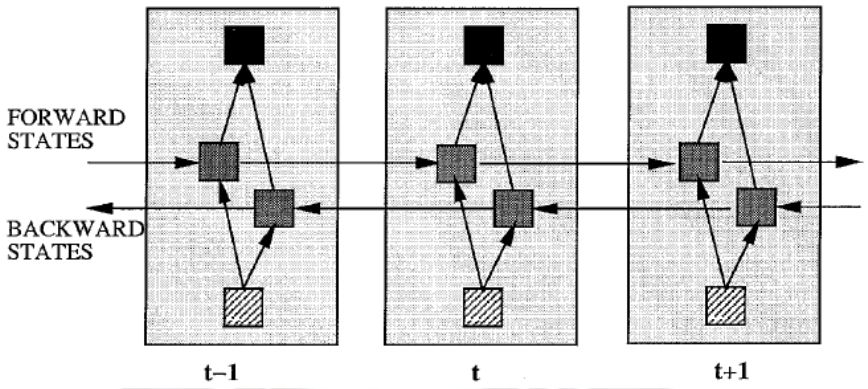


Figura 2.10. Arquitectura de una BRNN.
Fuente: “Bidirectional Recurrent Neural Networks”

CAPÍTULO 3

ESTADO DEL ARTE

En esta esta sección se describirán los trabajos de investigación más relevantes sobre los agentes conversacionales que detectan emociones y las incluyen en su dialogo. Asimismo, se describirán investigación que realizan la detección de emociones mediante el uso de texto y/o rostros. Las bases de datos consultadas fueron "*Socpus*", "*IEEEXplorer*" y "*Web of Science*", en la Tabla 3.1 se detallan las cadenas de búsqueda utilizadas para cada una de las búsquedas.

Tabla 3.1. Cadenas de búsqueda en base de datos

Base de datos	Cadena de búsqueda
Scopus	TITLE-ABS-KEY ((chatbot OR (conversation* AND agent)) AND (emotion* AND (detect* OR recogn* OR predict*)))
IEEEXplorer	(("Abstract":"chatbot" OR ("Abstract":"conversation*" AND "Abstract":"agent")) AND ("Abstract":"emotion*" AND ("Abstract":"detect*" OR "Abstract":"recogn*" OR "Abstract":"predict*")))
Web of Science	TS = ((chatbot OR (conversation* AND agent)) AND (emotion* AND (detect* OR recogn* OR predict*)))

3.1 Investigaciones sobre agentes conversacionales emocionales

En esta sección se describirán las publicaciones más recientes sobre algoritmos y arquitecturas utilizadas para el desarrollo de agentes conversacionales con capacidad de detección de emociones.

- La investigación realizada por Griol et al. (2019) presenta un agente conversacional (AC) que pueda interactuar el mundos virtuales (Second-Life). El agente es multimodal, representado por un avatar 3D y usa la voz como medio de comunicación. Asimismo, aplica técnicas de *Inteligencia Artificial* (AI), *Procesamiento Natural del Lenguaje* (NLP), *computación afectiva* y *Modelado de Usuarios*. El comportamiento del agente conversacional se encuentra controlado por un modelo estadístico entrenado con conversaciones. La arquitectura AC se presenta en Figura 3.1, asimismo el AC puede detectar tres (03) emociones (ira, aburrimiento y duda) y de acuerdo con el estado emocional cambia los mecanismos de respuesta.

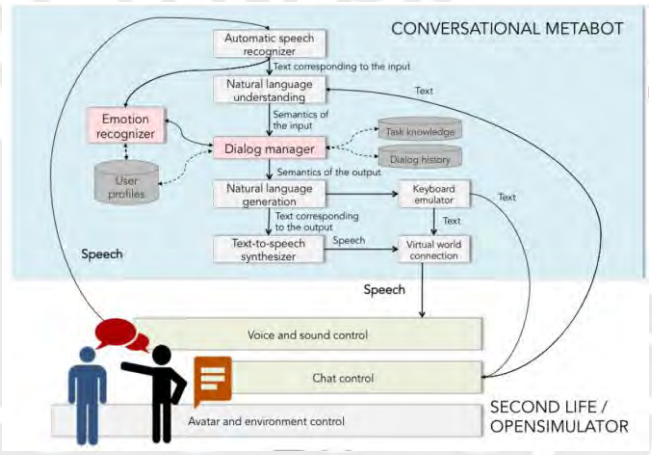


Figura 3.1. Arquitectura del AC para Second Life.
Fuente: *"Developing Enhanced Conversational Agents for Social Virtual Worlds"*

- CORK (Catania, Fisicaro, Spitale, & Garzotto, 2019) es un framework para el desarrollo de agentes conversacionales con inteligencia emocional y racional. El framework presentado fue probado en una plataforma virtual llamada '*Emoty*' para ayudar a personas con Alexitemia (incapacidad de reconocer las emociones propias) y para la recolección de información se usó un robot social '*Ele*' mediante la técnica *Wizard-of-Oz*. La arquitectura del framework es encuentra por distintos módulos como se muestra en la Figura 3.2, entre los más importantes: a) '*Engine*', que maneja todo el flujo y lógica del AC y llama otros módulos. b) '*Emotional Analysis*', reconociendo seis (06) emociones (alegría, tristeza, miedo, ira, disgusto, sorpresa); utiliza servicios web como el de 'IBM Tone Analyzer' ("Watson Tone Analyzer," n.d.) 'TheySay' ("Home | TheySay," n.d.) o 'Q°Emotion' ("Q°emotion - Home | Q°emotion - Emotional analysis

thanks to AI," n.d.). c) '*Output Creation*', que es el encargado de generar la respuesta basado en plantillas o ***Table-Driven***, la cual se selecciona de acuerdo con los resultados de los módulos (intención, tópico, contexto, emoción)

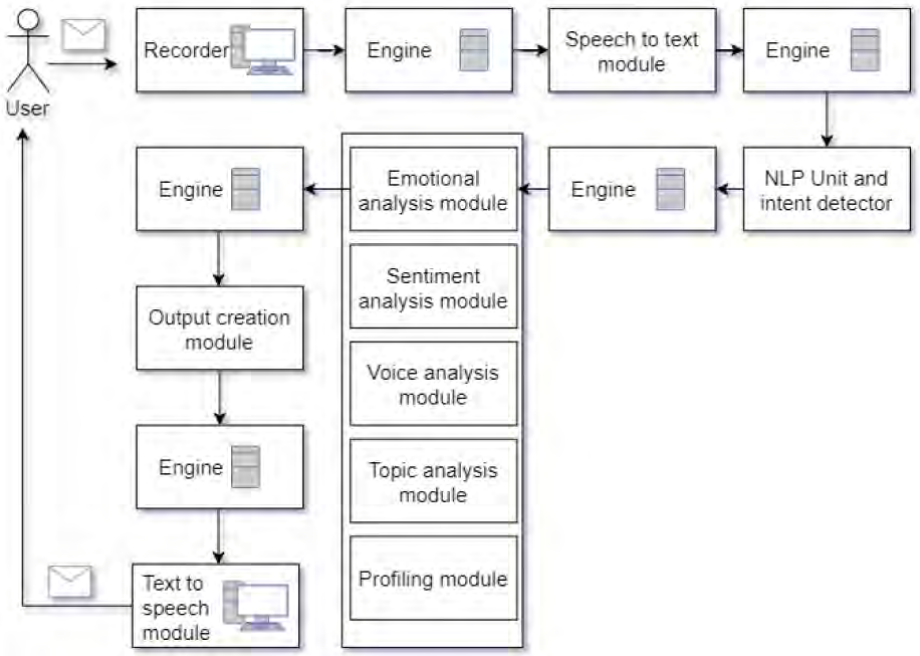


Figura 3.2. Flujo de procesamiento de CORK.
Fuente: *"CORK: A COnversational agent framewoRK exploiting both rational and emotional intelligence"*

- El artículo de C. Huang et al (2019) presenta un sistema de dialogo para expresar emociones, la metodología para entrenar el sistema es el siguiente: 1) Entrenar un modelo para la clasificación de emociones. 2) Etiquetar diálogos con el clasificador de emociones. 3) Entrenar un modelo de generación de respuestas con los diálogos etiquetados. El primer paso, entrena un LSTM Bidireccional (BiLSTM) con *self-attention* usando las bases de datos como el *CBET* (Gholipour Shahraki, 2015). En el segundo paso, etiqueta los diálogos obtenidos del *OpenSubtitles Corpus* (Lison & Tiedemann, 2016). Finalmente, entrena un modelo seq2seq con atención para la generación de respuestas, las emociones son incluidas en distintas parte del modelo como se muestra en la Figura 3.3, por ejemplo: antes del encoder (Enc-bef), después del encoder (Enc-aft), al inicio del decoder (Dec-start), entre otros. Asimismo, en orden de facilitar la evaluación, se utiliza el clasificador entrenado para determinar qué emoción es generada como respuesta; la métrica principal usada es la precisión o *accuracy*.

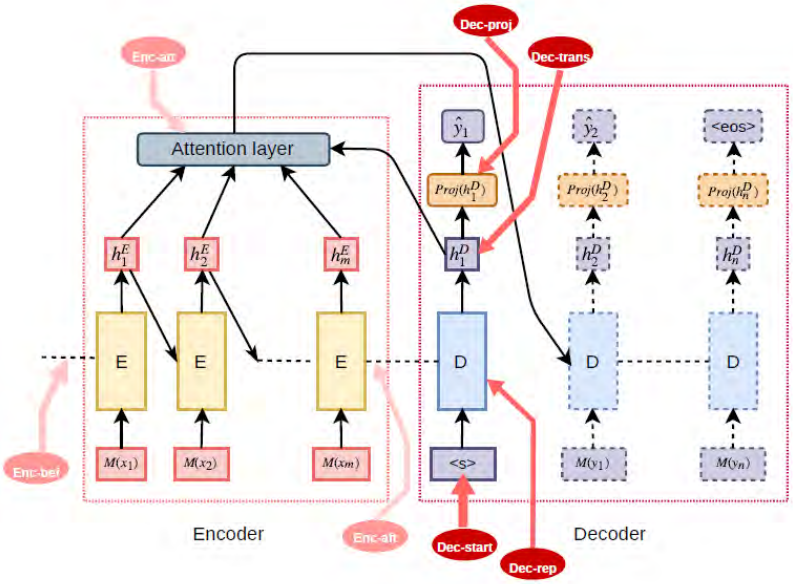


Figura 3.3. Posiciones del modelo seq2seq para incluir emociones.
Fuente:"*Generating Responses Expressing Emotion in an Open-domain Dialogue"*

- X. Sun et al (2018) describe un algoritmo red generativa antagónica (GAN, por sus siglas en inglés) usando el modelo seq2seq. Asimismo, menciona que el modelo tradicional del seq2seq puede generar problemas con frases cortas como "haha" o "good"; por este motivo sugiere la inclusión de un algoritmo de Reinforcement Learning (RL). Esta técnica permitirá asignar menores valores a las respuestas mal generadas por el algoritmo, en la Figura 3.4 y Figura 3.5 se muestra el generador y discriminador del modelo propuesto. Adicionalmente, tiene un clasificador de emociones basado en BiLSTM, que puede reconocer seis (06) emociones: ira, disgusto, felicidad, gusto, tristeza, otros.

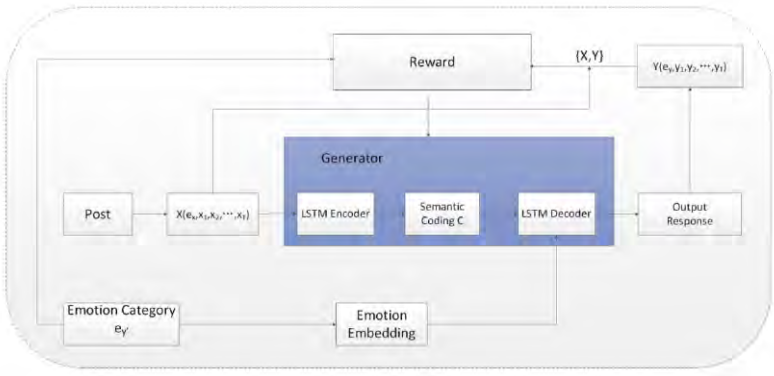


Figura 3.4. Generador del GAN con RL.
Fuente: "*Emotional Human Machine Conversation Generation Based on SeqGAN"*(Sun et al., 2018)

Figura 3.5. Discriminador de la GAN.
Fuente: *"Emotional Human Machine Conversation Generation Based on SeqGAN"*

- D. Lee et al (2017) propone un chatbot que puede generar respuestas emocionales mediante la continua observación del dialogo, el objetivo principal es ser usado para el área de salud mental. En la Figura 3.6 se puede ver como se realiza el procesamiento de una oración en las cuatro (04) etapas: 1) Extracción de características. 2) Decisión de Respuesta, que consiste en la intención de la oración (emoción y contexto). 3) Generación de respuesta informativa. 4) Generación de respuesta empática. Adicionalmente, agregan una restricción a la intención para saber si el usuario desea una respuesta emocional o informativa.

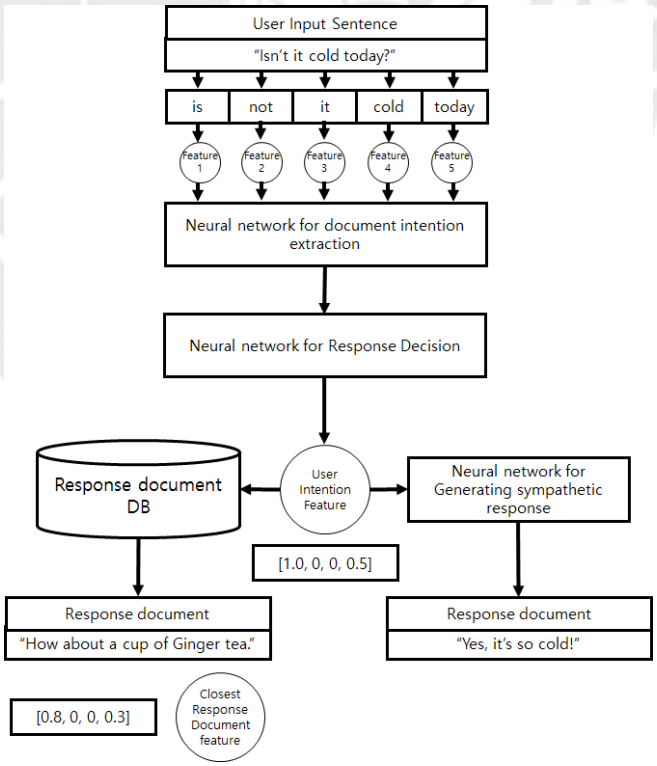


Figura 3.6. Proceso para la generación de respuesta.
Fuente: *"The Chatbot Feels You – A Counseling Service Using Emotional Response Generation"*

3.2 Investigaciones sobre la detección de emociones

En esta sección se describirán las publicaciones más recientes sobre algoritmos para la detección o clasificación de emociones mediante el uso de texto y/o expresiones faciales.

- Chatterjee et al. (2019) propone la identificación de cuatro (04) emociones: felicidad, tristeza, ira y otros, mediante el uso de Big-Data y Aprendizaje Profundo. El modelo propuesto se llama *Sentiment and Semantic Based Emotion Detector* (SS-BED), utiliza LSTM debido a que la información es secuencial (ver Figura 3.7). Asimismo, para la vectorización de las palabras usa dos (02) modelos de *Word-Embedding* que son: *Sentiment-Specific Word Embedding* (SSWE) y *Global Vector* (GloVe). La data utilizada fue recolectada mediante Twitter Firehose, una pequeña muestra de estas oraciones fue etiquetada y con un *embedding* seleccionaron las oraciones para cada emoción en base a la similaridad coseno. Finalmente, con toda la data etiquetada entrenaron el SS-BED donde usan la Entropía Cruzada con Softmax como función de costo.

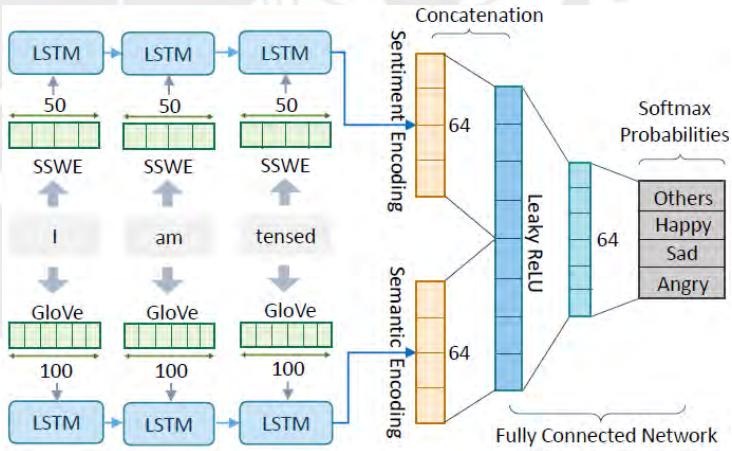


Figura 3.7. Representación del modelo SS-BED.
Fuente: *"Understanding emotions in text using deep learning and big data"*

- Dos (02) modelos supervisados para detección de emociones son propuestos por Mundra et al. (2017), el primero se llama *Emotion Detection using MinimalSupervised Emotion Lexico Generatio (ED-MSEL)* y el segundo *Emotion Detection using Neural Network Driven by Emotion Vector (ED-NNEV)*. No utilizan los dataset tradicionales, puesto que como su aplicación es para call-centers; las emociones detectadas son: felicidad, seguridad,

aprobación, cortesía, disculpa, desaprobación, no feliz y sin emoción. El modelo ED-MSEL se muestra en la Figura 3.8 y cuenta con una etapa de extracción de características, un lexicón de palabras por emociones generado en base de *Fuerza de Asociación* (SOA) e *Información Mutua* (PMI); finalmente se entrena un *Conditional Random Field* (CRF). El ED-NNEV usa la data etiquetada para crear un léxico, luego extrae palabras del corpus y crea vectores de emoción por cada palabra de la oración y se introducen a una red neuronal convolucional (CNN), en la Figura 3.9 se muestra la estructura del modelo.

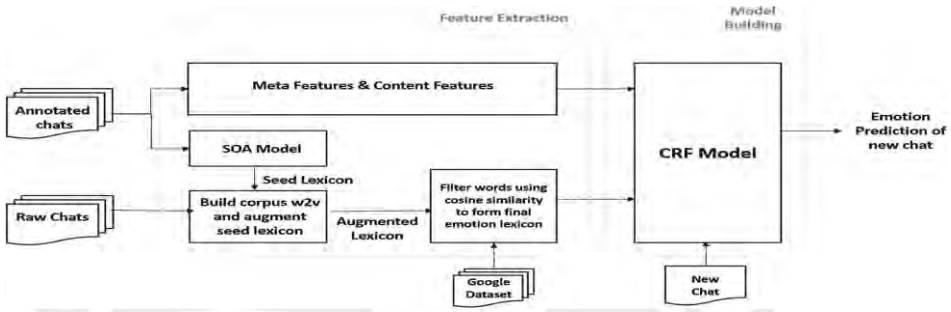


Figura 3.8. Representación del modelo ED-MSEL.
Fuente: *"Fine-Grained Emotion Detection in Contact Center Chat Utterances"*

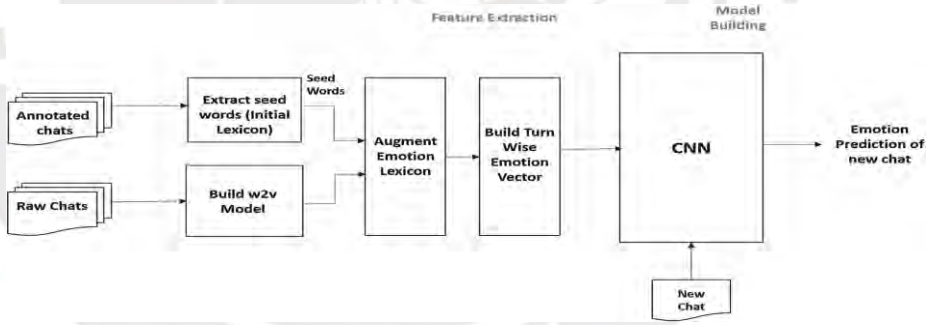


Figura 3.9. Representación del modelo ED-NNV.
Fuente: *"Fine-Grained Emotion Detection in Contact Center Chat Utterances"*

- El reconocimiento de emociones faciales mediante el uso de CNNs es propuesto por Dhankhar (2019), el algoritmo propuesto es el ensamble de las arquitecturas CNN llamadas ResNet50 y VGG-16. El ensamble es realizado mediante los dos vectores que generan las arquitecturas mencionadas y con estos entrenar una regresión logística para determinar las emociones; las bases de datos utilizadas son: *Kaggle's Facial Expession Recognition Challenge* y *Karolinska Directed Emotional Faces (KDEF)*.

- Depply-Supervised Neural Network (DSN) es propuesto por Fan et al. (2018) para el reconocimiento de emociones, el propósito es enriquecer los mapas de características producidos después de cada convolución de una CNN; para esto utilizan la deconvolución y el upsampling como se muestra en la Figura 3.10. Asimismo, redefinen las funciones de perdida para entrenar el modelo con los resultados intermedios progresivamente. Las CNNs utilizadas para este método son VGG-Face y ResNet50.

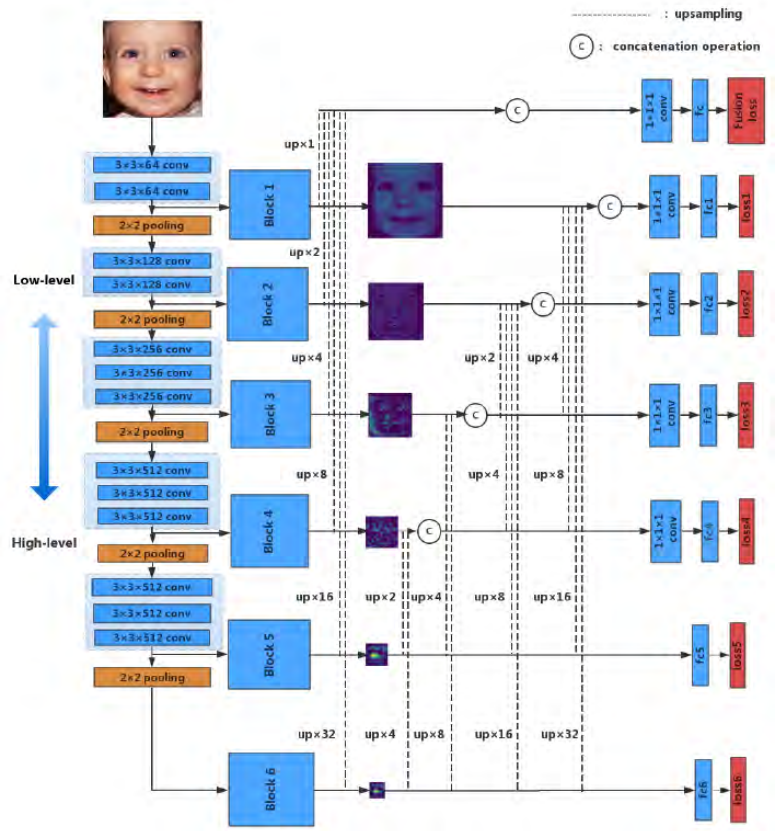


Figura 3.10. Arquitectura de la DSN.
Fuente: *"Video-based Emotion Recognition Using Deeply-Supervised Neural Networks"*

- El reconocimiento de emociones faciales también es usado en el campo de la robótica, el artículo de Mollahosseini et al. (2018) muestra la inclusión de la detección de emociones para mejorar el dialogo. El dataset utilizado es el *AffectNet* que contiene más de 1 millón de imágenes y contiene las siguientes emociones: neutral, feliz, triste, sorpresa, miedo, ira, disgusto, ninguna, no-rostro e incierto; la arquitectura entrenada es una CNN llamada ResNet.

CAPÍTULO 4

DISEÑO EXPERIMENTAL

En esta sección se realizará una descripción de los experimentos realizados para la elaboración de los modelos de detección de emociones para agentes conversacionales, Se consideran dos (02) modelos: 1) Clasificación de emociones en base a texto. 2) Clasificación de emociones en base a rostros.

Al no existir base de datos que tengan imágenes y texto simultaneo etiquetado como emociones, se asume una independencia en las fuentes de información cuando sean utilizados en los agentes conversacionales; por este motivo se entrenan los modelos por separado.

4.1 Modelo de clasificación de emociones en texto

La realización del modelo de clasificación de emociones se realizó en primer lugar con el uso de una (01) base de datos en inglés en donde se compararon diferentes soluciones basadas en RNN contra un modelo base que consiste en una regresión logística. Luego, se escogió el que mostraba una mejor performance en la tarea de clasificación.

4.1.1 Recolección de Base de Datos

a. Cleaned Balanced Emotion Tweets (CBET)

El CBET (Gholipour Shahraki, 2015) es una base de datos que contiene 81,163 Tweets etiquetados en nueve (09) emociones (ira, miedo, alegría, amor, tristeza, sorpresa, agradecimiento, disgusto y culpa) distintas de manera balanceada; asimismo

contiene algunas oraciones con múltiples etiquetas. En la Tabla 4.1 se muestra cuantas oraciones corresponden a cada emoción y en la Tabla 4.2 cuantas oraciones contienen múltiples etiquetas. Se observa que las emociones estas balanceadas con respecto al total de tweets y que solo el 5.3% de las oraciones son multi-etiquetadas,

Tabla 4.1. Distribución de emociones en el CBET.
Fuente: Propia

Emoción	Número de Tweets
Ira	9,073
Miedo	9,110
Alegría	10,889
Amor	11,640
Tristeza	9,275
Sorpresa	9,287
Agradecimiento	8,919
Disgusto	8,647
Culpa	8,626

Tabla 4.2. Cantidad de tweets con etiquetas múltiples en el CBET.
Fuente: Propia

Número de etiquetas	Número de Tweets	Porcentaje
1	76,860	94.7 %
2	4,303	5.3 %

4.1.2 Pre-procesamiento

En esta etapa se realizará el pre-procesamiento de las bases de datos antes mencionadas, para esto se realizarán los siguientes pasos:

1. Convertir a minúsculas toda la oración.
2. Eliminar todos los emoticones y emojis.
3. Eliminar todos los signos de puntuación y los hashtags (#) de los tweets.
4. Eliminar todos los correos y usuarios que empiezan con arroba (@).

El proceso de pre-procesamiento se realizó utilizando dos (02) librerías de Python. La primera es '*re*' (Python Software Foundation, n.d.) que nos permite generar expresiones regulares en texto; y la segunda es '*gensim*' (Rehurek & Sojka, 2010) que es un framework para el modelado de tópicos y cuenta con herramientas para el Procesamiento Natural del Lenguajes como modelos de vectorización y '*stopwords*'. En la Tabla 4.3 se muestra un ejemplo de las oraciones antes y después de realizar los pasos 1-4.

Tabla 4.3. Oraciones después del pre-procesamiento.
Fuente: Propia

Oración Inicial	Oración Procesada
THE #PARTY IS HERE tonight ! #BYOB #SHOW first PARTY after major rager #MAJORRAGER halloween #HALLOWEEN	the party is here tonight byob show first party after major rager majorrager halloween halloween
Creates #Wealth #Success 😀	creates wealth success

4.1.3 Vectorización de palabras

La vectorización de las palabras son representaciones que permiten encontrar similaridad entre palabras comunes ("What Are Word Embeddings for Text?," n.d.); son modelos no-supervisados que permiten capturar la similaridad semántica y usualmente entrenados con mucho texto (Rostylsav Neskorozhenyi, n.d.). La vectorización de palabras de las oraciones se realizó utilizando dos (02) tipos de modelos: 'word2vec' y 'FastText'.

El modelo 'Word2Vec' fue desarrollado por Mikolov en (2013) donde propone dos (02) modelos para reducir la complejidad; asimismo, su estudio demuestra que se pueden entrenar vectores de mayor dimensión con largas bases de datos para conseguir mayores relaciones semánticas como '*ciudad-país*'. El segundo modelo, 'FastText' presentado por Bojanowski et al. (2017) describe un modelo basado en '*skipgram'* donde cada palabra es una bolsa de caracteres '*n-gram'*. Los vectores en inglés se obtuvieron con la librería '*gensim'* que permite entrenar los modelos con distintos corpus que se encuentran incluido y variar la dimensión de los vectores, la dimensión de 300 fue seleccionada para estandarizar con los modelos pre-entrenados en español utilizados. Los vectores de palabra se utilizaron para realizar la tokenización de oraciones; se realizaron dos (02) formas de tokenización:

- Tokenización de la oración completa, para ser aplicado en el modelo base de Regresión Logística. En este modelo se crean vectores oración mediante el uso SWEM-*aver* (Shen et al., 2018), esta aproximación consiste en promediar todos los vectores que se encuentren en la oración; una modificación fue eliminar de este promedio a los vectores que no se encuentran en los modelos de vectores de palabra.

- Tokenización de la oración con número limitado de palabras para los modelos basados en RNN. Se escogió el tamaño de 25 palabras, que es 25% más de las palabras que se recomiendan al escribir en el idioma inglés, donde una guía general da un rango de 15-20 palabras (Cutts, 2013).

La tokenización se realizó que el diccionario obtenido por el corpus '*text8*' de la librería '*gensim*', este diccionario no cuenta con todas las palabras por lo que se creó un token '<UNK>' que reemplaza a las palabras desconocidas por el diccionario. Luego, de realizar las tokenizacion se eliminaron oraciones donde más del 50% de sus palabras fueran desconocidas o la oración no tuviera ninguna palabra, reduciéndose la cantidad de oraciones de 76,860 a 76,575. Finalmente, se incluye el '*padding*' en caso de oraciones muy cortas o largas con el token '<PAD>'.

4.1.4 Modelos de clasificación

Las bases de datos de oraciones con etiquetas múltiples de emociones se dividen en (03), el 80% de la información es utilizada para el entrenamiento, 10% para la afinación de parámetros y otros 10% para las pruebas de los modelos; la división se realizó con la librería '*scikit-learn*' (Scikit-learn, n.d.; Varoquaux et al., 2015). En cada modelo se probó el dataset completo con las nueve (09) emociones y el dataset solo con las seis (06) emociones de Ekman, con el objetivo de comprobar que al reducir la complejidad del problema mejora el comportamiento del modelo. A continuación, se describen los (03) modelos comparados para la detección de emociones en texto:

a. **Modelo 1:** Regresión Logística

   Este modelo se realiza como base de comparación, consta de seis (06) modelos entrenados independientemente para cada emoción. Las características para la predicción, es la concatenación de los vectores oración producidos por el 'Word2vec' y el 'FastText' de manera independiente; lo cual genera un vector de características de una dimensión de 600. La salida de cada modelos es del tipo binario (0/1) y representa si la emoción está o no en la oración. Usando varios modelos individuales se crea un vector de multi-etiquetas el cuál será evaluado posteriormente. El modelo de regresión logística se realizó con '*scikit-learn*' con las siguientes características: C=1.0, class_weight='balanced' y max_iter=2000.

b. **Modelo 2:** LSTM

Este modelo desarrollado se basa en el presentado por Chatterjee et al. (2019) en el que usa dos (02) LSTM, cada uno vectorizado con distintas modelos distintos de *Word-Embeddings*. En el modelo propuesto se utiliza FastText y Word2Vec como los modelo de embeddings; cada LSTM independientemente genera un vector de 32 características que se concatenan y pasan por una capa LeakyReLu y una Softmax. En la siguiente figura se puede apreciar la estructura del modelo desarrollado con LSTM.

Figura 4.1. Modelo LSTM utilizado para detectar emociones en texto.
Fuente: Propia

c. **Modelo 3**: Bi-LSTM (LSTM Bidireccional)

Este modelo utiliza la misma arquitectura descrita para el modelo con la diferencia de que las LSTM son reemplazadas por Bi-LSTM, y genera un vector de 64 en veces de las 32 características de la LSTM.

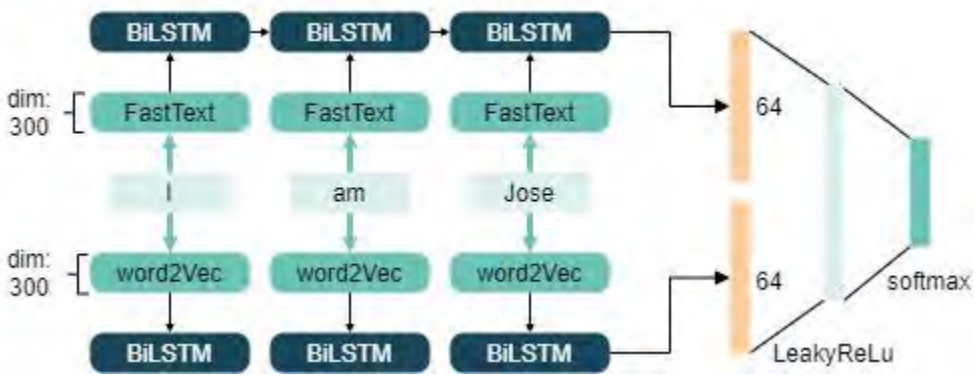


Figura 4.2. Modelo Bi-LSTM utilizado para detectar emociones en texto.
Fuente: Propia

4.2 Modelo de clasificación de emociones en rostros

El modelo de clasificación de emociones en rostros de realizo utilizando la base de datos 'Affectnet' (Mollahosseini, Hasani, & Mahoor, 2019), se entrenaron tres (03) modelos basados en CNN y usando el concepto de Transfer Learning para reducir el tiempo de procesamiento. Luego, se compararon los modelos para escoger el que mejor se desempeñaba con el menor costo computacional.

4.2.1 Recolección de Base de Datos

a. AffecNet

El AffecNet (Mollahosseini et al., 2019), (Mahoor, n.d.) es una base de datos creada en la Universidad de Denver y reúne aproximadamente 1 millón de imágenes, de las cuales 440 mil han sido etiquetadas manualmente. El dataset cuenta con una clasificación discreta y continua de las emociones usando dos (02) modelos emocionales. En la Tabla 4.4 se muestra la distribución de las emociones en el AffectNet de las imágenes etiquetadas manualmente.

Tabla 4.4. Distribución de emociones en el AffectNet
Fuente: Propia

Emoción	Número de Muestras
Neutral	75,374
Alegría	134,915
Tristeza	25,959
Sorpresa	14,590
Miedo	6,878
Disgusto	4,303
Ira	25,382
Desprecio	4,250
Ninguna	33,588
Incierto	12,145
No-Rostro	82,915

Debido a la gran cantidad de imágenes en el AffecNet, se realizaron modificaciones para reducir la cantidad de muestras y quedarnos con las clases de interés, a continuación, se listan las tres (03) modificaciones realizadas:

- Se combinan las emociones 'Disgusto' y 'Desprecio'; que de acuerdo al modelo de emociones de Ekman pertenecen a una misma categoría.

- Solo se seleccionan las seis (06) emociones correspondientes al modelo de emociones de Ekman.
- Solo se seleccionaron como máximo 8,600 muestras por cada emoción con el objetivo de balancear la base de datos.

Luego, de realizar las modificaciones mencionadas el dataset se redujo de 420,299 a 49,831; esta reducción permitió poder reducir el tiempo de entrenamiento de los modelos.

4.2.2 Modelos de clasificación

Las imágenes etiquetas múltiples de emociones se dividen en (03), el 80% de la información es utilizada para el entrenamiento, 10% para la afinación de parámetros y otros 10% para las pruebas de los modelos. En cada modelo se probó el dataset solo con las seis (06) emociones de Ekman, las imágenes se pre-procesan redimensionándolas al tamaño 48x48 y 150x150 para estandarizar la entrada a los tres (03) modelos y luego evaluar la influencia del tamaño de la imagen en el rendimiento. Asimismo, cada imagen es modelo pre-entrenado (VGG16 y ResNet50) cuentan con su función de pre-procesamiento previa a la predicción de características. A continuación, se describen los (03) modelos comparados para la detección de emociones en rostros:

a. **Modelo 1:** VGG16

   El modelo VGG16 (Simonyan & Zisserman, 2015) realizado utiliza los pesos pre-entrenados de las capas convolucionales de la arquitectura que vienen en la librería '*Tensorflow* 2.0' (TensorFlow, n.d.) y que han sido entrenados con el dataset ImageNet de la Universidad de Princeton. Las capas convolucionales son utilizadas para generar un vector carácteristico de dimensión 18,432; con esto se entrena una red neuronal de dos (02) capas con 256 neuronas ReLu, Dropout de 0.3 y una capa Softmax.

b. **Modelo 2:** ResNet50

   El modelo ResNet50 (He et al., 2016) desarrollado sigue las mismas características que el modelo 1 descrito anteriormente, con la diferencia que en las capas convolucionales se agrega un AveragePooling2D para obtener un vector característico de dimensión 18,432.

c. **Modelo 3:** miniXception

El tercer modelo se basa en la arquitectura planteada por O. Arraiga (2017), en donde propone la eliminación de las capas finales que aumentan el costo computacional. El modelo hace uso de cuatro (04) módulos residuales y convoluciones *Depth-wise Separable*, asimismo realiza un GlobalAvergePooling2D final junto a una capa Softmax con la cantidad de clases que se requieran. En la Figura 4.3 se puede observar el modelo descrito.

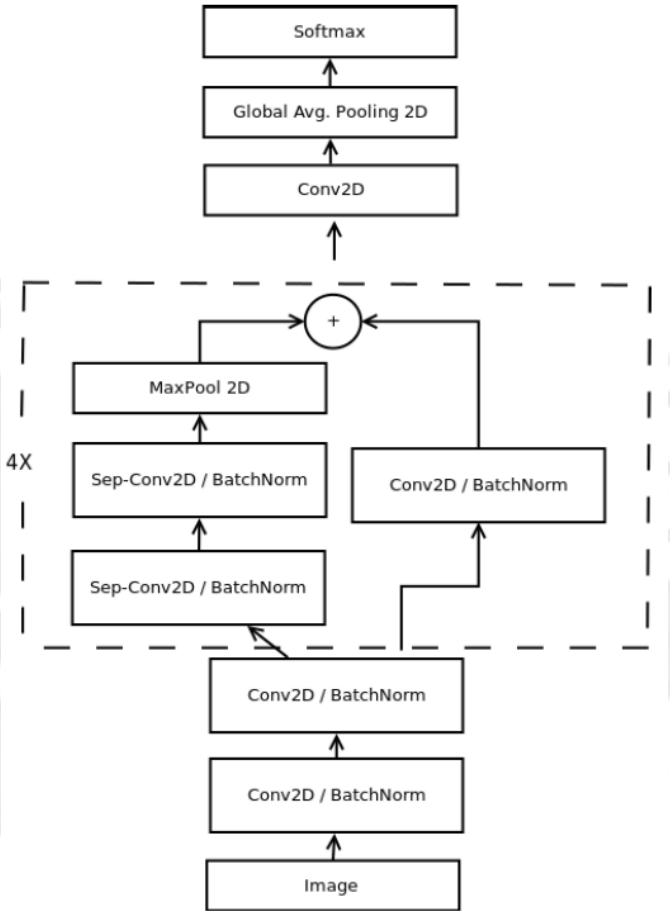


Figura 4.3. Modelo miniXception.
Fuente: "Real-time Convolutional Neural Networks for Emotion and Gender Classification"

CAPÍTULO 5

RESULTADOS Y DISCUSIÓN

En esta sección se presentarán los resultados de los modelos aplicados y que fueron descritos en el capítulo anterior, se evaluaran las métricas obtenidas en los conjuntos de datos de pruebas y se comparan los resultados de todos los modelos desarrollados con el objetivo de seleccionar los mejores para ser utilizados por un agente conversacional.

5.1 Resultados de clasificación de emociones en texto

La evaluación de los modelos entrenados se realiza mediante el uso de la matriz de confusión multi-clase, asimismo se calculan para cada clase las métricas 'Accuracy', 'Precision', 'Recall' y 'F1-Score'; asimismo el macro-averaged de cada una de las métricas mencionadas. Luego, se compararán las macro-averaged de cada modelo para seleccionar el que mejor se comporta para la tarea de clasificar emociones en texto.

5.1.1 Modelo Regresión Logística

El entrenamiento del modelo de Regresión Logística (RL) se realizó con las características anteriormente mencionadas usando el conjunto correspondiente, el conjunto de validación se utilizó para comprar métricas y verificar que los resultados mejoraban al reducir la complejidad del problema, es decir disminuir el número de clases a identificar por el modelo.

a. Modelo entrenado con 9 emociones

En este caso luego del entrenamiento de la regresión logística para las nueve (09) emociones del dataset CBET se obtiene la matriz de confusión que se muestra en la Figura 5.1 y las métricas mostradas en la Tabla 5.1. Se puede observar que las emociones con el mayor número de Verdaderos Positivos y mejores métricas son: "*Miedo*", "*Agradecimiento*", "*Amor*" y "*Disgusto*"; dos de estas clases no se encuentran incluidas en las emociones de Ekman.

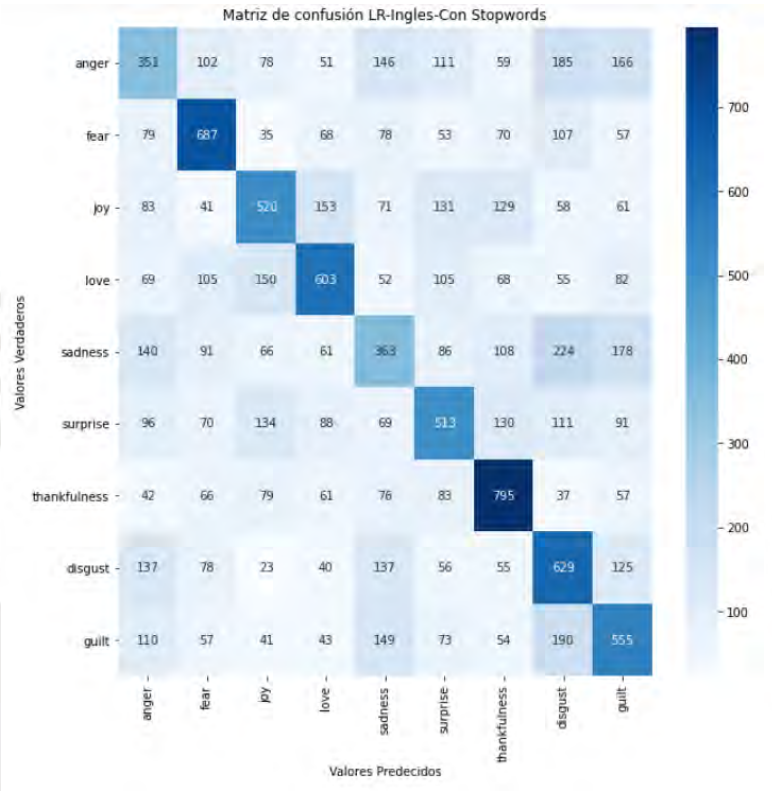


Figura 5.1. Matriz de Confusión multi-clase para nueve (09) emociones - RL.
Fuente: Propia

Tabla 5.1. Métricas modelo RL – Con 9 emociones
Fuente: Propia

Emoción	Accuracy	Precision	Recall	F1-Score
Ira	85.59%	31.71%	28.10%	29.79%
Miedo	**89.93%**	**52.96%**	**55.67%**	**54.29%**
Alegría	88.39%	46.18%	41.70%	43.82%
Amor	**89.11%**	**51.63%**	46.78%	**49.08%**
Tristeza	84.92%	31.81%	27.56%	29.54%
Sorpresa	87.05%	42.36%	39.40%	40.83%
Agradecimiento	**89.78%**	**54.16%**	**61.34%**	**57.53%**
Disgusto	85.91%	39.41%	**49.14%**	43.74%
Culpa	86.65%	40.45%	46.32%	41.98%
	87.48%	**43.41%**	**43.70%**	**43.40%**

b. Modelo entrenado con 6 emociones

En este caso luego del entrenamiento de la regresión logística para el CBET modificado para seis (06) emociones se obtiene la matriz de confusión que se muestra en la Figura 5.2 y las métricas mostradas en la Tabla 5.2. Se puede observar que las emociones con el mayor número de Verdaderos Positivos y mejores métricas son: "*Miedo*" y "*Alegría*". Asimismo, a pesar de que disminuye el *accuracy* general el resto de las métricas mejoran, así como la cantidad de verdaderos positivos por clase.

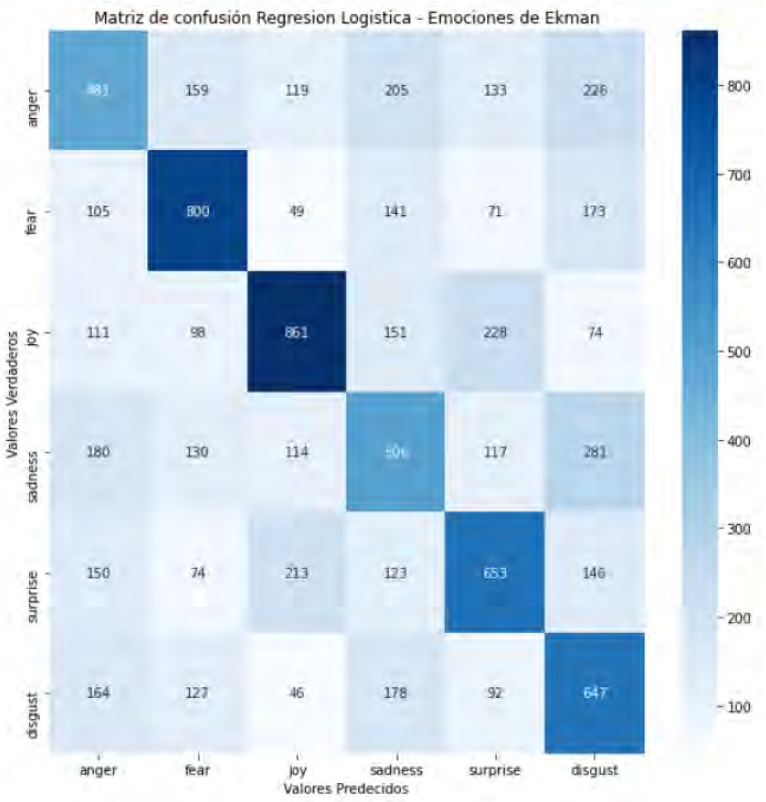


Figura 5.2. Matriz de Confusión multi-clase para seis (06) emociones - RL.
Fuente: Propia

Tabla 5.2. Métricas modelo RL – Con 6 emociones
Fuente: Propia

Emoción	Accuracy	Precision	Recall	F1-Score
Ira	80.90%	40.39%	36.36%	38.26%
Miedo	**86.13%**	**57.64%**	**59.75%**	**58.67%**
Alegría	**85.19%**	**61.41%**	**56.53%**	**58.87%**
Tristeza	80.06%	38.80%	38.10%	38.45%
Sorpresa	83.42%	50.47%	48.05%	49.23%
Disgusto	81.46%	41.82%	54.95%	46.19%
	82.86%	**48.42%**	**48.39%**	**48.28%**

5.1.2 Modelo LSTM

El entrenamiento del modelo de LSTM se realizó con por diez (10) épocas, debido al costo computacional y porque se observó que las gráficas de entrenamiento se mantenían constantes. Asimismo, se usó un Batch-Size de 128, las LSTM cuentan con Dropout de 0.5. El optimizador usado fue 'Adam' y la función perdida 'Entropía Cruzada Categorica'.

a. Modelo entrenado con 9 emociones

Luego del entrenamiento se obtiene la matriz de confusión que se muestra en la Figura 5.3 y las métricas mostradas en la Tabla 5.3. Se puede observar que las emociones con el mayor número de Verdaderos Positivos y mejores métricas son las mismas que en la Regresión Logística.

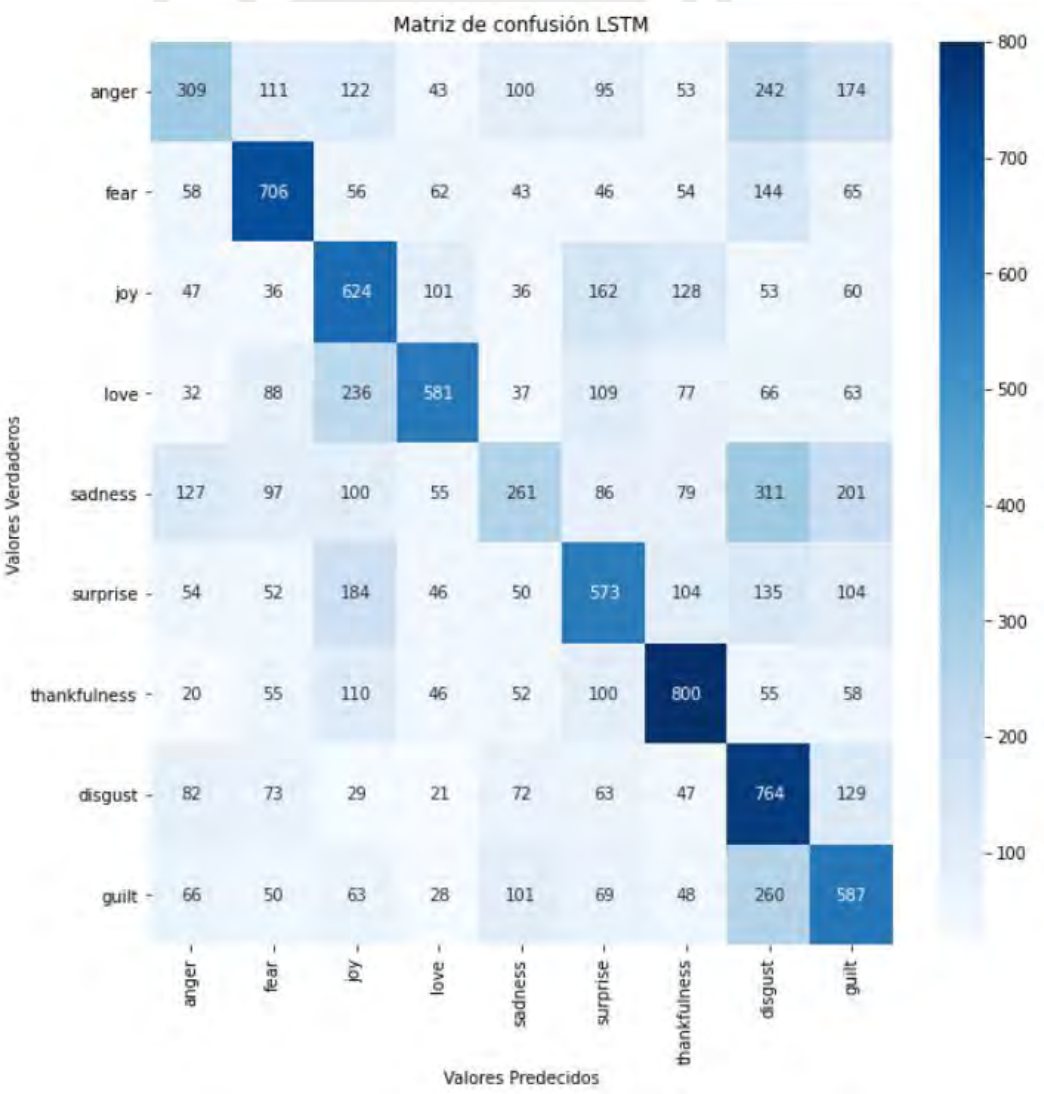


Figura 5.3. Matriz de Confusión multi-clase para nueve (09) emociones - LSTM.
Fuente: Propia

Tabla 5.3. Métricas modelo LSTM – Con 9 emociones
Fuente: Propia

Emoción	Accuracy	Precision	Recall	F1-Score
Ira	87.58%	38.87%	24.74%	30.23%
Miedo	**90.51%**	**55.68%**	**57.21%**	**56.43%**
Alegría	86.74%	40.94%	50.04%	45.04%
Amor	**90.34%**	**59.10%**	45.07%	**51.14%**
Tristeza	86.53%	34.71%	19.82%	25.23%
Sorpresa	87.30%	43.98%	44.01%	43.99%
Agradecimiento	**90.55%**	**57.55%**	**61.73%**	**59.57%**
Disgusto	84.49%	37.64%	**59.69%**	46.16%
Culpa	86.60%	40.74%	46.15%	43.27%
	87.85%	**45.46%**	**45.38%**	**44.56%**

b. Modelo entrenado con 6 emociones

Luego del entrenamiento se obtiene la matriz de confusión que se muestra en la Figura 5.4 y las métricas mostradas en la Tabla 5.4. Se puede observar el mismo comportamiento que en la Regresión Logística, con diferencia que en la métrica de '*recall'* la clase Disgusto tiene el más alto valor.

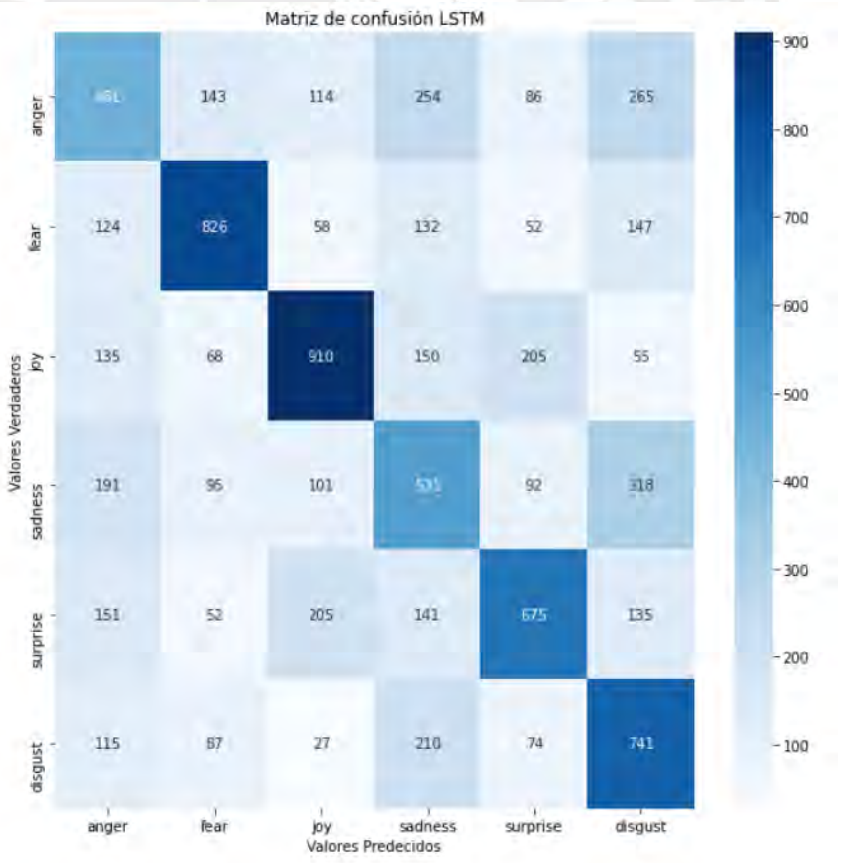


Figura 5.4. Matriz de Confusión multi-clase para seis (06) emociones - LSTM.
Fuente: Propia

Tabla 5.4. Métricas modelo LSTM – Con 6 emociones
Fuente: Propia

Emoción	Accuracy	Precision	Recall	F1-Score
Ira	80.58%	39.17%	34.85%	36.88%
Miedo	88.21%	64.99%	61.69%	63.30%
Alegría	86.24%	64.31%	59.75%	61.95%
Tristeza	79.28%	37.45%	39.98%	38.67%
Sorpresa	85.32%	57.01%	49.67%	53.09%
Disgusto	82.37%	44.61%	59.09%	50.84%
	83.66%	**51.26%**	**50.84%**	**50.79%**

5.1.3 Modelo Bi-LSTM

El entrenamiento del modelo de Bi-LSTM se realizó con por diez (10) épocas, debido al costo computacional y porque se observó que las gráficas de entrenamiento se mantenían constantes. Asimismo, se usó un Batch-Size de 128, las LSTM que forma la red bidireccional cuentan con Dropout de 0.5. El optimizador usado fue '*Adam'* y la función perdida '*Entropía Cruzada Categorica'*.

a. Modelo entrenado con 9 emociones

Luego del entrenamiento se obtiene la matriz de confusión que se muestra en la Figura 5.5 y las métricas mostradas en la Tabla 5.5. Se puede observar que las emociones con el mayor número de Verdaderos Positivos y mejores métricas son las mismas que el entrenamiento que con la LSTM.

Tabla 5.5. Métricas modelo Bi-LSTM – Con 9 emociones
Fuente: Propia

Emoción	Accuracy	Precision	Recall	F1-Score
Ira	87.16%	36.98%	25.70%	30.33%
Miedo	**91.02%**	**58.42%**	**57.05%**	**57.73%**
Alegría	87.86%	44.54%	48.04%	46.22%
Amor	**89.21%**	**51.88%**	53.45%	**52.66%**
Tristeza	86.32%	34.24%	20.96%	26.00%
Sorpresa	88.69%	50.14%	42.47%	45.99%
Agradecimiento	**90.99%**	**59.52%**	**62.96%**	**61.19%**
Disgusto	85.64%	40.00%	**57.66%**	47.23%
Culpa	85.92%	39.32%	49.92%	43.99%
	88.09%	**46.12%**	**46.47%**	**45.70%**

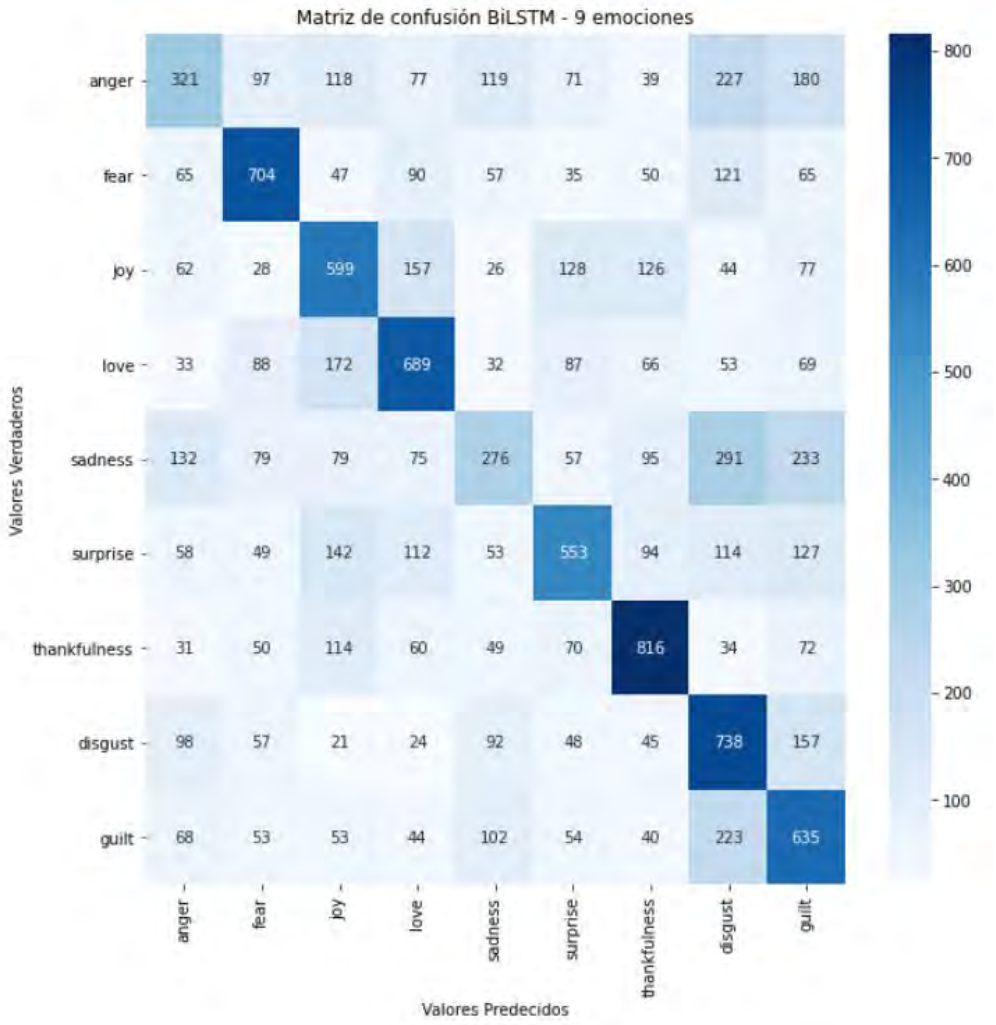


Figura 5.5. Matriz de Confusión multi-clase para nueve (09) emociones – Bi-LSTM.
Fuente: Propia

b. Modelo entrenado con 6 emociones

Luego del entrenamiento se obtiene la matriz de confusión que se muestra en la Figura 5.6 y las métricas mostradas en la Tabla 5.6. Se puede observar el mismo comportamiento que en la Regresión Logística.

Tabla 5.6. Métricas modelo Bi-LSTM – Con 6 emociones
Fuente: Propia

Emoción	Accuracy	Precision	Recall	F1-Score
Ira	81.91%	42.99%	34.09%	38.02%
Miedo	**89.60%**	**71.22%**	**61.91%**	**66.24%**
Alegría	**84.85%**	**58.46%**	**66.25%**	**62.11%**
Tristeza	80.88%	41.20%	39.68%	40.43%
Sorpresa	84.27%	53.04%	52.02%	52.53%
Disgusto	83.44%	47.11%	59.17%	52.54%
	84.16%	**52.34%**	**52.19%**	**51.97%**

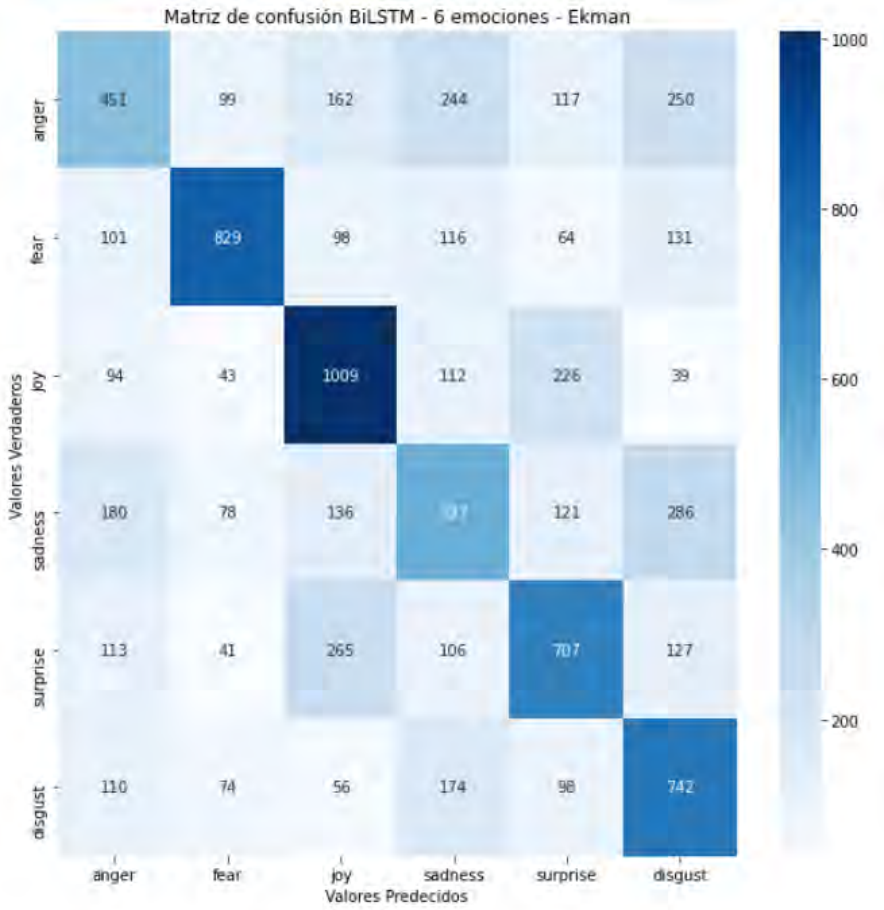


Figura 5.6. Matriz de Confusión multi-clase para seis (06) emociones – Bi-LSTM.
Fuente: Propia

5.1.4 Comparación de Modelos

En las siguientes tablas se muestra la comparación de las métricas de los distintos modelos entrenados para la clasificación de nueve y seis emociones respectivamente, de las tablas se observa lo siguiente:

- En primer lugar, que al reducir el número de clases del problema el '*macro-averaged accuracy'* disminuye, no obstante, esta disminución mejora las demás métricas aumentando sin importar el modelo un promedio de 5% en cada métrica.
- El segundo punto a resaltar es que los modelos LSTM y Bi-LSTM se comportan mucho mejor que la Regresión Logística, y aunque la diferencia en el '*accuracy'* no es resaltante; en el resto de las métricas la diferencia es entre 2-3%.

- El tercer punto observable, es que la diferencia de métricas entre la LSTM y el Bi-LSTM es menor al 1%, pero el número de parámetros adicionales necesarios por el Bi-LSTM para ese mejoramiento es aproximadamente un 50% mayor. Por lo que el modelo LSTM podría ser el mejor en caso la cantidad de oraciones que procesa por unidad de tiempo sea mayor que el modelo Bi-LSTM.
- Finalmente, con los modelos entrenados para nueve (09) emociones se puede observar que mediante el uso de un procesamiento simple del texto y un diccionario de palabras genérico se obtuvo mejores resultados que el SVM entrenado por el creador del CBET usando el Lexicon NRC.

Tabla 5.7. Comparación de Métricas Macro-Averaged – Con 9 emociones
Fuente: Propia

Modelo	Nro Parametros	Accuracy	Precision	Recall	F1-Score
SVM[1]	-	-	45.01%	42.26%	43.59%
Regresión Logística	-	87.48%	43.41%	43.70%	43.40%
LSTM	85,833	87.85%	45.46%	45.38%	44.56%
Bi-LSTM	129,033	**88.09%**	**46.12%**	**46.47%**	**45.70%**

Tabla 5.8. Comparación de Métricas Macro-Averaged – Con 6 emociones
Fuente: Propia

Modelo	Nro Parametros	Accuracy	Precision	Recall	F1-Score
Regresión Logística	-	82.86%	48.42%	48.39%	48.28%
LSTM	85,638	83.66%	51.26%	50.84%	50.79%
Bi-LSTM	128,646	**84.16%**	**52.34%**	**52.19%**	**51.97%**

5.2 Resultados de clasificación de emociones en rostros

Las imágenes utilizadas para el entrenamiento de los tres (03) modelos se organizaron con una estructura de carpetas de acuerdo a lo recomendado en la documentación de '*Tensorflow 2.0*' para la utilización de las imágenes desde carpetas (TensorFlow, n.d.-b), en la siguiente imagen se observa la estructura de archivos.

[1] El modelo SVM es el desarrollado por el creador del CBET en la tesis "*Emotion Mining from Text*" en este modelo A. Gholipour utiliza un Lexicon NRC (https://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm) para entrenar su modelo.

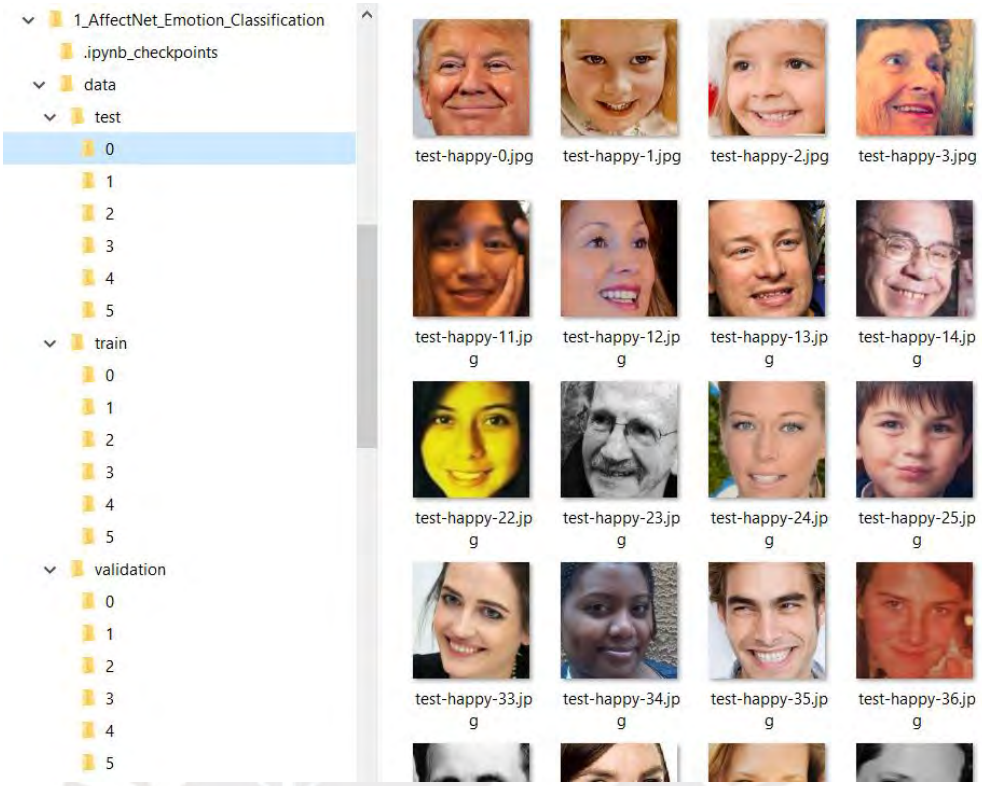


Figura 5.7. Estructura de carpetas para el AffectNet.
Fuente: Propia

Asimismo, al igual que con la clasificación de emociones en texto se usaran los mismos instrumentos de medición: 1) Matriz de confusión multi-clase. 2) Métricas *Macro-averaged*; estas métricas se evalaurán en imágenes de distintos tamaños 48x48 y 150x150 pixeles. Adicionalmente, se tendrá en consideración la cantidad de parámetros que utilizan las CNN, puesto que se evalaura su influencia en el tiempo de inferencia.

5.2.1 Modelo VGG16

El modelo VGG16 utilizado tiene un total de 14’714,688 millones de parámetros correspondientes a las capas convolucionales. La red neuronal se entrenó con un Batch-size de 64 y durante 20 épocas. Asimismo, se utilizó una función de parada temprana y una de reducción de la tasa de aprendizaje; estas funciones nos permiten mejorar el entrenamiento de la red, así como evitar que la red siga entrenando si no existe una mejora en la función de perdida y/o la métrica de evaluación. La matriz de confusión para los resultados con imágenes de 48x48 se muestra en la Figura 5.8 y las métricas en la Tabla 5.9; y de imágenes de 150x150 en la Figura 5.9 y Tabla 5.10.

Tabla 5.9. Métricas VGG16 – Imágenes 48x48
Fuente: Propia

Emoción	Accuracy	Precision	Recall	F1-Score
Alegría	**78.85%**	**41.54%**	**46.22%**	**43.76%**
Tristeza	74.71%	28.14%	28.85%	28.49%
Sorpresa	76.66%	34.52%	**42.52%**	38.11%
Miedo	**85.41%**	**42.51%**	32.16%	36.62%
Disgusto	78.47%	28.96%	20.58%	24.06%
Ira	77.14%	38.13%	41.55%	**39.77%**
	78.54%	**35.63%**	**35.31%**	**35.13%**

Tabla 5.10. Métricas VGG16 – Imágenes 150x150
Fuente: Propia

Emoción	Accuracy	Precision	Recall	F1-Score
Alegría	**85.15%**	**57.76%**	**61.67%**	**59.65%**
Tristeza	69.68%	30.35%	**56.90%**	39.58%
Sorpresa	78.85%	40.48%	53.56%	**46.11%**
Miedo	**87.68%**	**62.26%**	15.16%	24.38%
Disgusto	79.73%	34.03%	23.73%	27.96%
Ira	82.84%	54.50%	33.48%	41.48%
	80.65%	**56.56%**	**40.75%**	**39.86%**

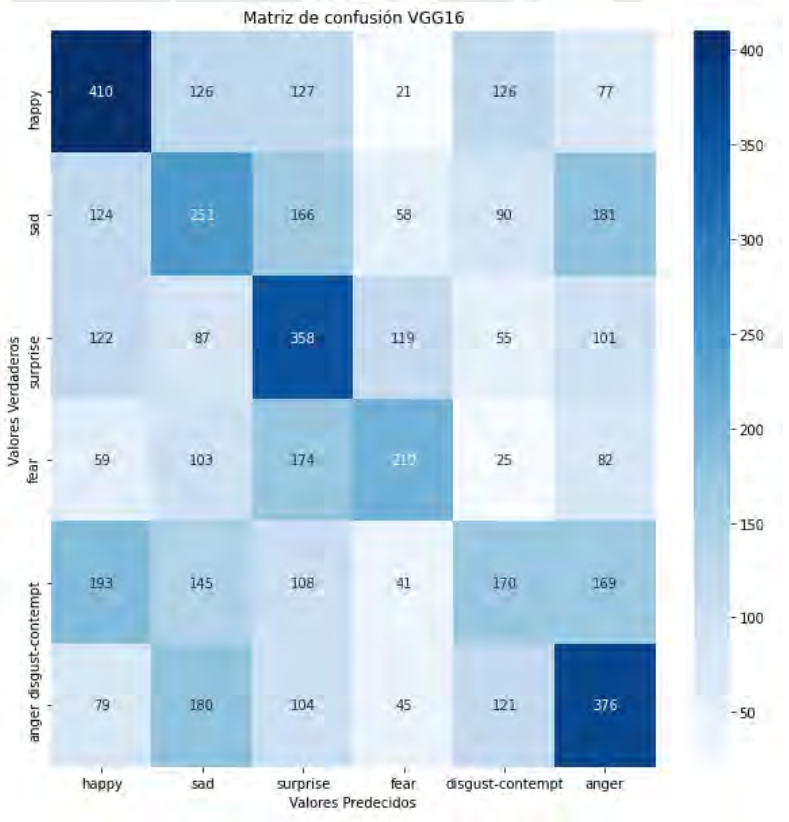


Figura 5.8. Matriz de confusión multi-clase usando VGG16 – Imágenes 48x48.
Fuente: Propia

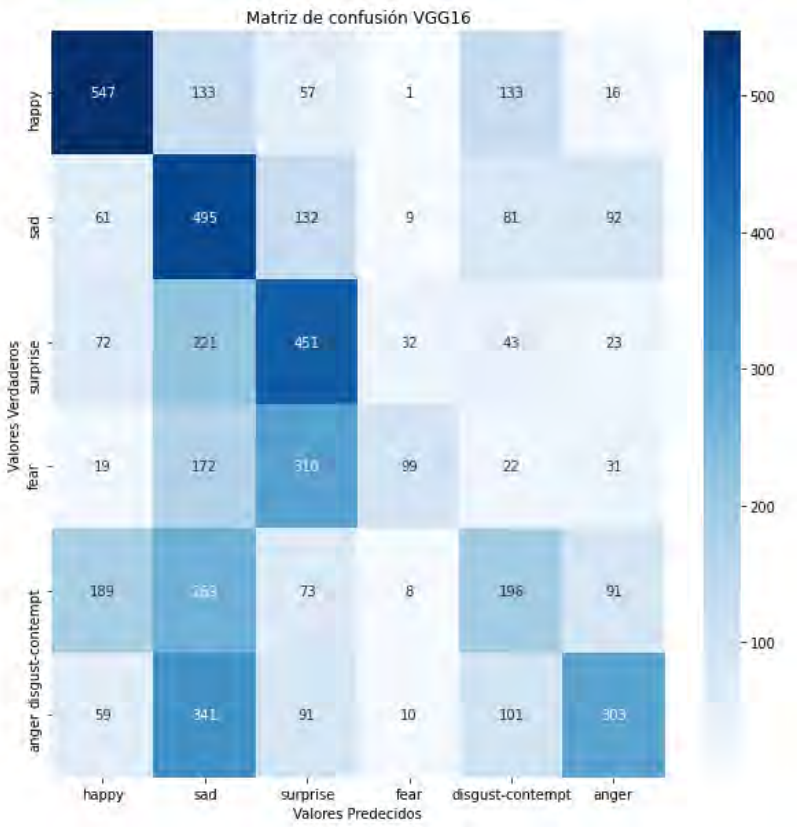


Figura 5.9. Matriz de confusión multi-clase usando VGG16 – Imágenes 150x150.
Fuente: Propia

De las métricas obtenidas se puede observar que las clases que tienen más de una métrica sobresaliente son: 'Alegría' y 'Miedo'. Asimismo, se puede observar que cuando se incrementa el tamaño de la imagen las métricas macro-averaged aumentan considerablemente, especialmente la '*Precision'* aumenta un 21% aproximadamente; y el resto de métricas entre un 2-5%. Analizando la matriz de confusión de imágenes de 150x150, se observa que el modelo predice de manera incorrecta es '*Tristeza*', y la confunde con 'Ira', 'Disgusto' y 'Sorpresa'. Otras dos emociones que se confunden bastante son 'Alegría' con ´Disgusto', 'Miedo' con 'Ira', y 'Sorpresa' con 'Miedo'

5.2.2 Modelo ResNet50

El modelo ResNet50 utilizado tiene un total de 23'587,712 millones de parámetros correspondientes a las capas convolucionales. La red neuronal se entrenó con un Batch-size de 64 y durante 20 épocas. Asimismo, se utilizó las mismas funciones auxiliares que en el VGG16. Las matrices de confusión para los dos tamaños de imágenes se muestran en la Figura 5.10 y Figura 5.11; y las métricas en la Tabla 5.11 y Tabla 5.12.

Tabla 5.11. Métricas ResNet50 – Imágenes 48x48
Fuente: Propia

Emoción	Accuracy	Precision	Recall	F1-Score
Alegría	**82.56%**	**51.10%**	**47.13%**	**49.03%**
Tristeza	76.36%	33.62%	36.32%	34.92%
Sorpresa	75.82%	35.42%	**52.38%**	**42.26%**
Miedo	**86.17%**	**44.97%**	24.66%	31.85%
Disgusto	77.20%	29.33%	26.63%	27.92%
Ira	79.07%	42.09%	40.55%	41.31%
	79.53%	**39.42%**	**37.94%**	**37.88%**

Tabla 5.12. Métricas ResNet50 – Imágenes 150x150
Fuente: Propia

Emoción	Accuracy	Precision	Recall	F1-Score
Alegría	**89.06%**	**72.32%**	**62.46%**	**67.03%**
Tristeza	78.71%	40.86%	49.08%	44.60%
Sorpresa	82.28%	48.01%	**58.67%**	**52.81%**
Miedo	**87.82%**	**55.61%**	34.92%	42.90%
Disgusto	78.23%	36.72%	43.34%	39.76%
Ira	82.92%	53.56%	44.86%	48.83%
	83.17%	**51.18%**	**48.89%**	**49.31%**

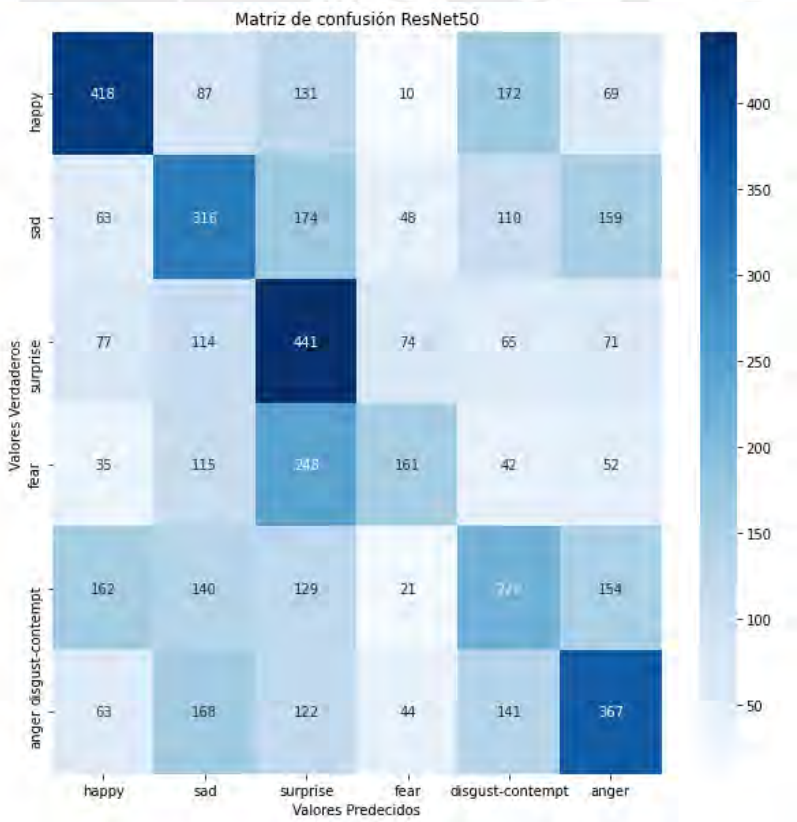


Figura 5.10. Matriz de confusión multi-clase usando ResNet50 – Imágenes 48x48.
Fuente: Propia

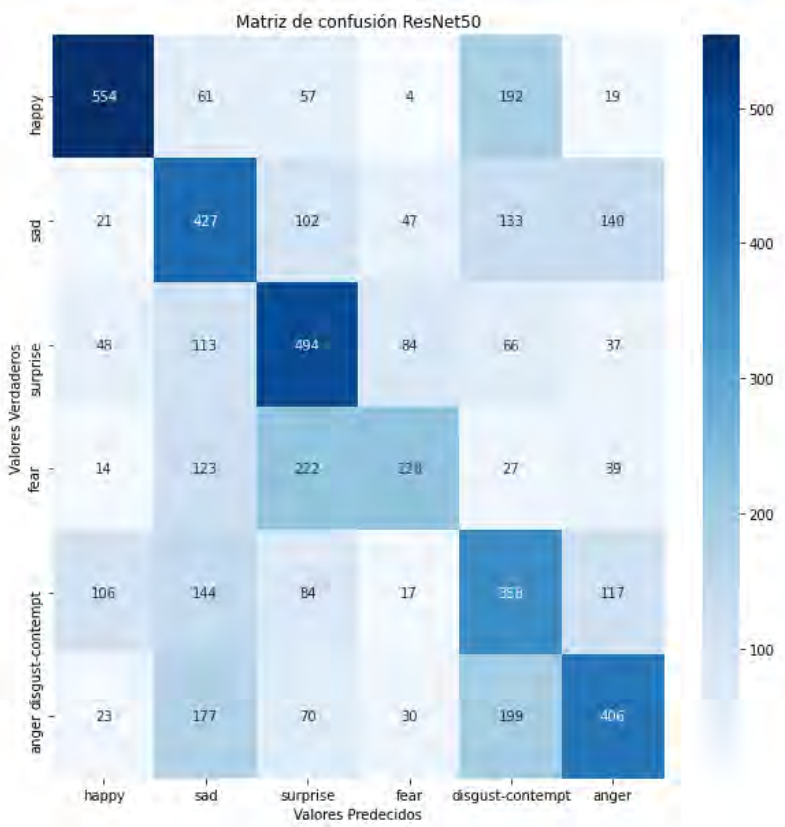


Figura 5.11. Matriz de confusión multi-clase usando ResNet50 – Imágenes 150x150.
Fuente: Propia

Al igual que el modelo VGG16 la clase con mejores métricas es ‘Alegría’ y ‘Miedo’, no obstante el modelo usando ResNet50 tiene métricas con mayores valores; esto también se comprueba en la matriz de confusión con una mayor cantidad de verdaderos positivos. Adicionalmente, se puede visualizar que algunas confusiones entre clases se mantienen y otras se hacen más evidentes como el caso de ´Miedo’ con ‘Sorpresa’. Que las confusiones se repitan son muestra que la complejidad para distinguir entre estas clases es elevada, y se analizara sin se mantienen en otros modelos.

5.2.3 Modelo miniXception

El modelo miniXception utilizado tiene solo 57,414 miles de parámetros correspondientes a las capas convolucionales. La red neuronal se entrenó con un Batch-size de 128 y durante 20 épocas. Asimismo, se utilizó las mismas funciones auxiliares que en los modelos anteriores. Las matrices de confusión para cada tamaño de imagen se muestran en la Figura 5.12 y la Figura 5.13; sus respectivas métricas en la Tabla 5.13 y Tabla 5.14.

Tabla 5.13. Métricas miniXception – Imágenes 48x48
Fuente: Propia

Emoción	Accuracy	Precision	Recall	F1-Score
Alegría	**88.14%**	**65.81%**	**69.45%**	**67.58%**
Tristeza	81.12%	46.15%	48.97%	47.52%
Sorpresa	83.32%	50.61%	54.51%	52.49%
Miedo	**86.41%**	47.92%	42.27%	44.91%
Disgusto	80.23%	39.15%	34.75%	36.82%
Ira	83.72%	**55.18%**	**55.36%**	**55.27%**
	83.83%	**50.80%**	**50.88%**	**50.77%**

Tabla 5.14. Métricas miniXception – Imágenes 150x150.
Fuente: Propia

Emoción	Accuracy	Precision	Recall	F1-Score
Alegría	**89.97%**	**71.57%**	**72.38%**	**71.97%**
Tristeza	84.85%	56.12%	60.57%	58.26%
Sorpresa	86.39%	**59.43%**	**61.40%**	**60.40%**
Miedo	**88.80%**	57.83%	53.75%	55.71%
Disgusto	81.72%	44.47%	41.40%	42.88%
Ira	84.95%	58.60%	58.34%	58.47%
	86.11%	**58.00%**	**57.98%**	**57.95%**

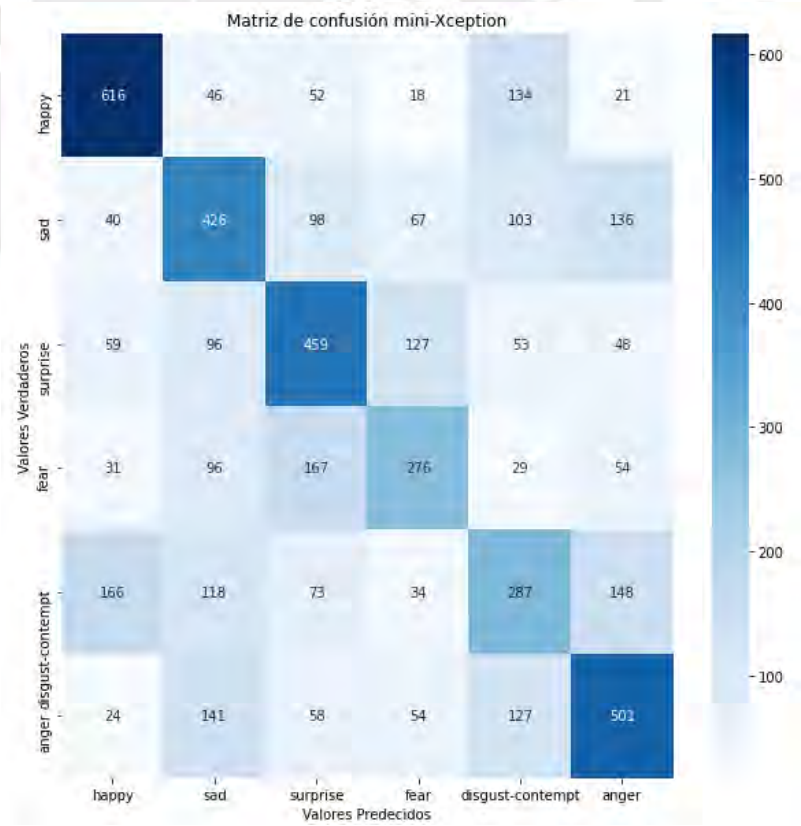


Figura 5.12. Matriz de confusión multi-clase usando miniXception – Imágenes 48x48.
Fuente: Propia

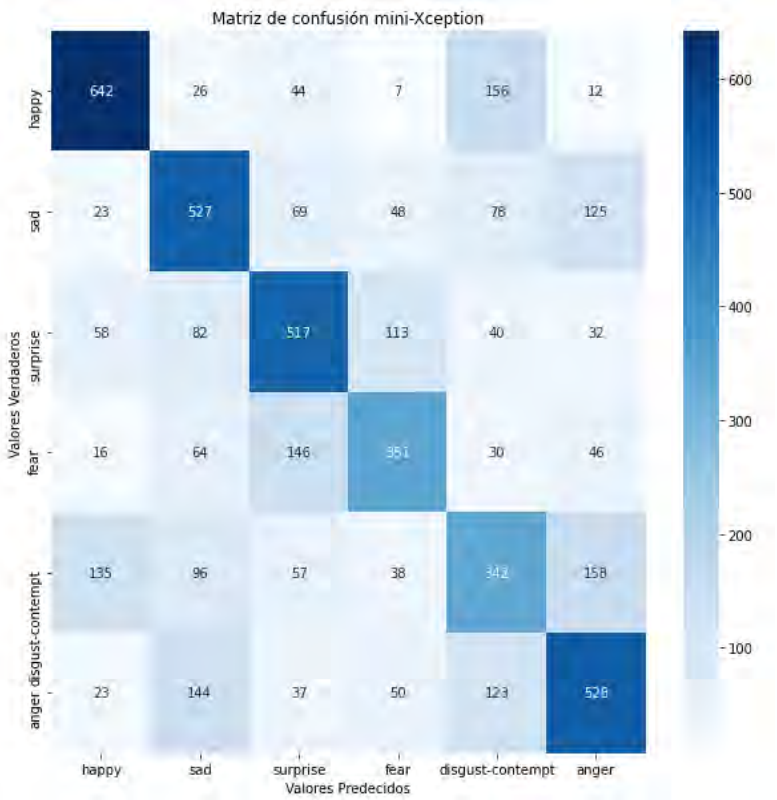


Figura 5.13. Matriz de confusión multi-clase usando miniXception – Imágenes 150x150.
Fuente: Propia

En el modelo miniXception se puede ver que la clase mejor clasificada es 'Alegría' como en los modelos anteriores, por otro lado la segunda mejor clase cambia de acuerdo al tamaño de la imagen. En el caso de 48x48, es 'Ira' y para 150x150 es 'Sorpresa'. Las confusiones de clases se mantienen igual que en los modelos anteriores, lo que puede significar se necesita más muestras o en caso contrario es un problema muy complejo para las CNN analizadas.

5.2.4 Comparación de modelos

Finalizado, el entrenamiento de los tres (03) modelos para detección de emociones en rostros se realizó una comparación que se muestra en las tablas Tabla 5.15 y Tabla 5.16, en estas se observa lo siguiente:

- El modelo que demuestra un mejor comportamiento es el miniXception con imágenes de 150x150, alcanzando el mayor 'Accuracy' entre los tres modelos con un valor de 86.1%
- Se observa que el modelo miniXception tiene mejor 'Recall' en todos los modelos analizados, aproximadamente 15-17%; esta métrica nos indica que el modelo ha clasificado mejor la emociones con respecto a lo real.

- Por último, modelo miniXception utiliza menor cantidad de parámetros significando menor número de cálculos al momento de realizar una predicción. En distintos artículos (Canziani, Paszke, & Culurciello, 2016; Velasco-Montero, Fernández-Berni, Carmona-Galán, & Rodríguez-Vázquez, 2018) se ha comprobado mediante pruebas en sistemas de bajo poder computacional (ej. Rasberry Pi 3, Jetson TX1) que a menor número de parámetros la predicción se realiza en menor tiempo y la cantidad de frame-per –second (fps) incrementa. Esto será probado y analizado en la siguiente sección.

Tabla 5.15. Comparación de Métricas Macro-Averaged – AffectNet Imágenes 48x48
Fuente: Propia

Modelo	Nro Par. Body (CNN)	Nro. Par. Head (NN)	Accuracy	Precision	Recall	F1-Score
VGG16	14'714,688	198,662	78.54%	35.63%	35.31%	35.13%
ResNet50	23'587,712	591,878	79.53%	39.42%	37.94%	37.88%
miniXception	57,414	0	**83.83%**	**50.80%**	**50.88%**	**50.77%**

Tabla 5.16. Comparación de Métricas Macro-Averaged – AffectNet Imágenes 150x150
Fuente: Propia

Modelo	Nro Par. Body (CNN)	Nro. Par. Head (NN)	Accuracy	Precision	Recall	F1-Score
VGG16	14'714,688	2'164,742	80.65%	56.56%	40.75%	39.86%
ResNet50	23'587,712	2'164,742	83.17%	51.18%	48.89%	49.31%
miniXception	57,414	0	**86.11%**	**58.00%**	**57.98%**	**57.95%**

5.3 Resultados de pruebas de rendimiento

En esta sección se analizará principalmente el '*Throughput*' de los modelos implementados, en el caso de los clasificadores de texto se analizará cuantas oraciones por segundo puede procesar los modelos LSTM y Bi-LSTM. Por otro lado, en el caso de las imágenes se analizará cuantas imágenes puede clasificar por segundo. Las características del hardware donde se realizó el análisis son:

- **Procesador:** Intel Core i7-5700HQ CPU @ 2.7GHz
- **Sistema Operativo:** Windows 10 Home – 64bits
- **RAM:** 16 GB
- **GPU:** NVIDIA GTX 960M – 2GB

5.3.1 Modelos de detección de emociones en texto

En este caso se utilizaron oraciones de distintas longitudes de manera discreta (ej. 3, 5, 10 y 15 palabras), estas oraciones se seleccionaron del CBET con el cuál se entrenaron los modelos. Asimismo, la emoción predicha para dicha oración es realizada distinto número de veces para analizar como varia el rendimiento de acuerdo a la cantidad de oraciones a las que se somete el modelo. En esta tarea la librería '*time*' de Python nos permitió saber calcular el tiempo promedio de inferencias de los distintos modelos.

En la Figura 5.14 se muestra la gráfica de cuantas oraciones por segundo puede procesar el modelo LSTM para distinta cantidad de oraciones y longitudes de oraciones. Se puede observar que para repeticiones cortas (1, 5, 10, 25 y 50) la cantidad de palabras influye en el rendimiento del modelo. Por otro lado, al aumentar la cantidad de repeticiones el rendimiento tiende a ser el mismo sin importar la longitud de la oración y tienden a un valor aproximado de 15.5 – 16.0 oraciones por segundo. En la Figura 5.15, se muestra el comportamiento de la Bi-LSTM que presenta un comportamiento parecido al modelo LSTM, y para repeticiones grandes tiende a un valor aproximado entre 14.5 – 15.0 oraciones por segundo.

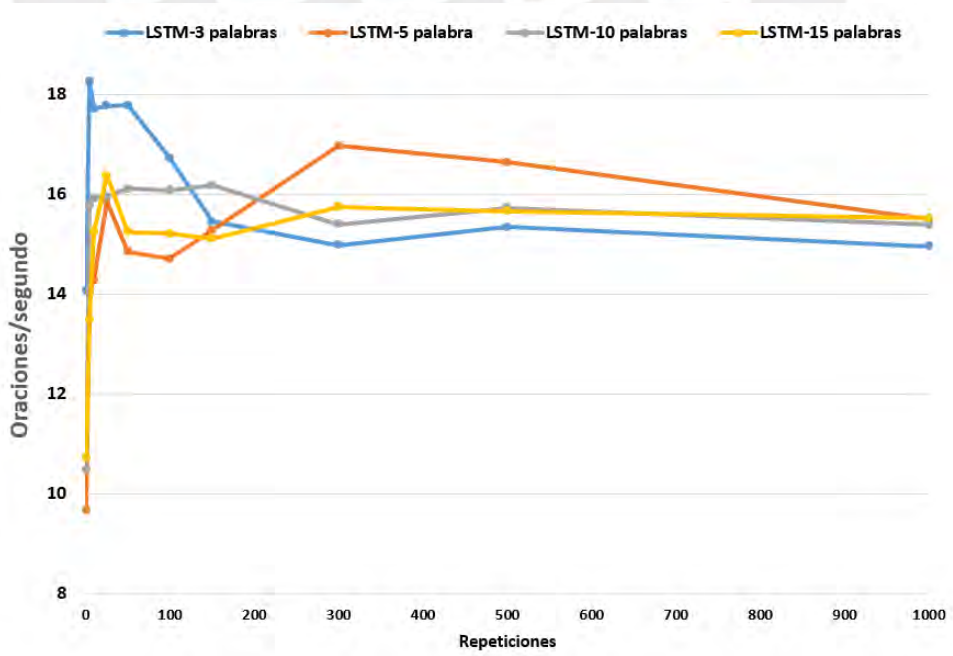


Figura 5.14. Gráfico de '*Throughput'* del LSTM.
Fuente: Propia

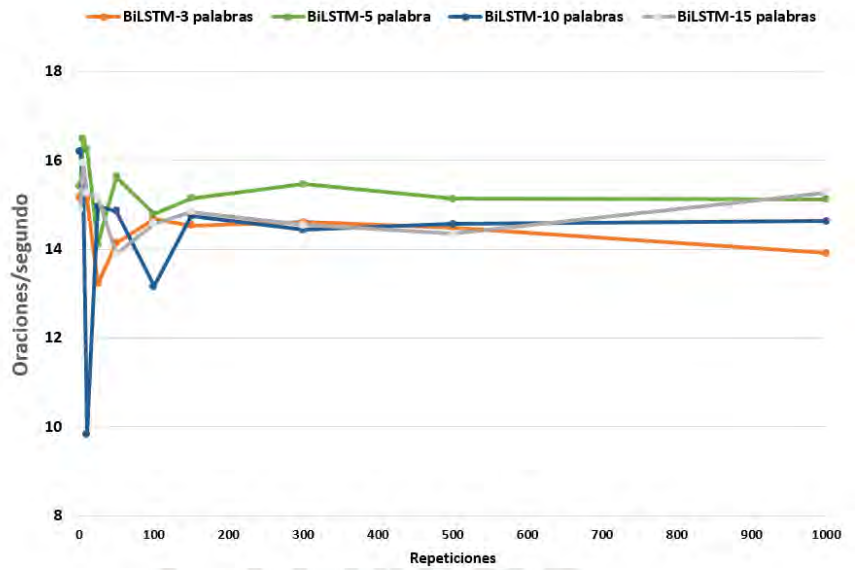


Figura 5.15. Gráfico de '*Throughput'* del BiLSTM.
Fuente: Propia

5.3.2 Modelos de detección de emociones en rostros

El análisis realizado para verificar el rendimiento en tiempo real de la detección de emociones en rostros se realizó usando una imagen aleatoria del dataset, creando un batch de 1, que simularía el funcionamiento de detectar 1 imagen por vez del modelo. La prueba solo mide el tiempo de predicción, no el pre-procesamiento de la imagen como el cambio tamaño. En la Figura 5.16 se muestra la gráfica para los modelos con imágenes de 48x48 y en la Figura 5.17 para imágenes de 150x150.

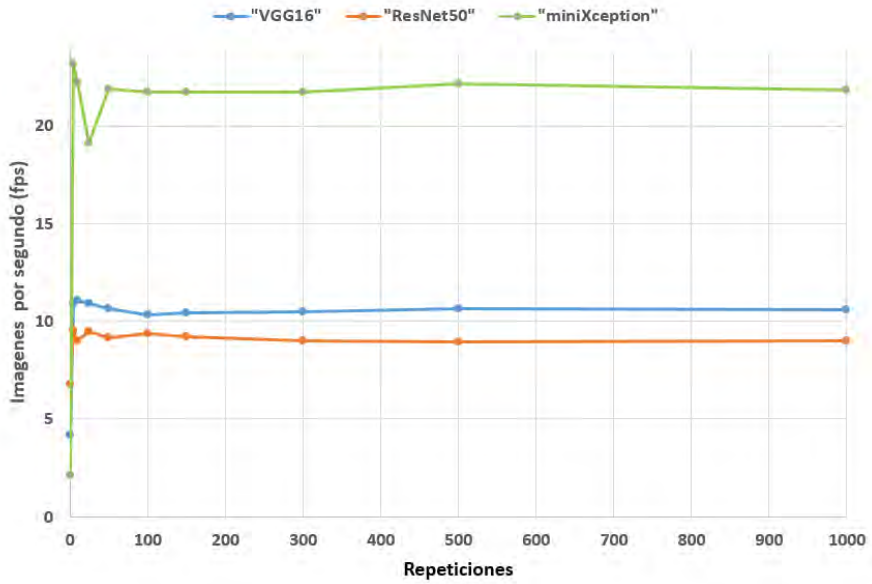


Figura 5.16. Gráfico de Imágenes por segundo (*fps*) de los modelos de detección de emociones – Imágenes 48x48.
Fuente: Propia

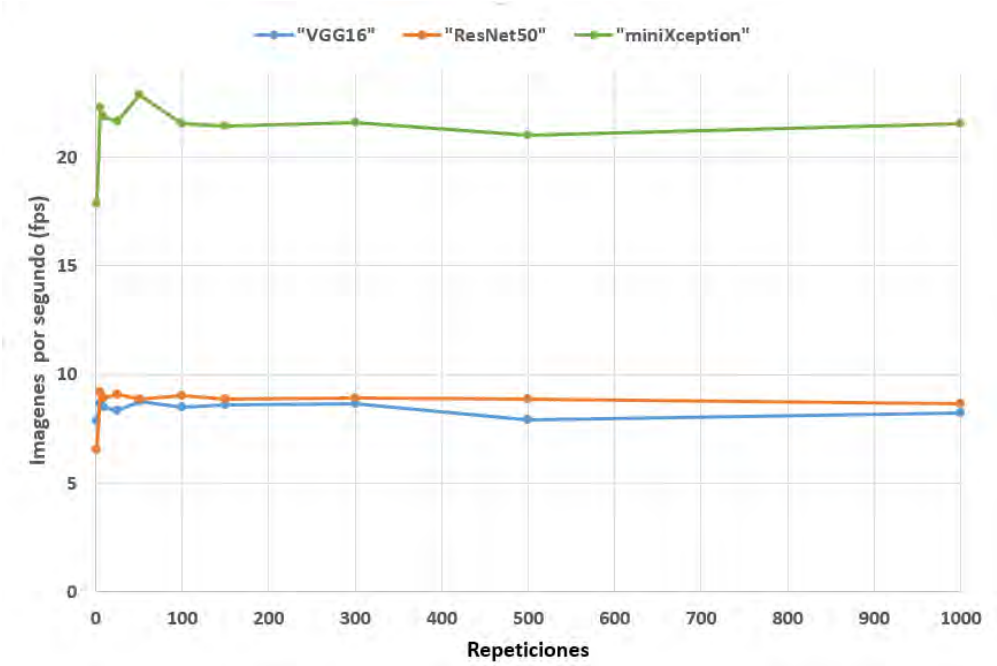


Figura 5.17. Gráfico de Imágenes por segundo (*fps*) de los modelos de detección de emociones - Imágenes 150x150.
Fuente: Propia

Se puede observar que en ambos casos el modelo miniXception tiene un rendimiento promedio alrededor de 20-21 fps para las imágenes de distintos tamaños, el cuál es aproximadamente el doble de los casos del VGG16 y ResNet50 que en el mejor de los casos llegan 9-10 fps. En base a esta comparación se puede comprobar que a menor número de parámetros el modelo se comportará mejor y su rendimiento aumentará en tiempo real; es por esto que el miniXception al tener solo miles de parámetros procesa más imágenes por segundo.

CONCLUSIONES

Al culminar la evaluación y análisis de los modelos para detección de emociones en texto y en rostros que puedan desempeñarse en tiempo real, se puede concluir lo siguiente:

- Primero, en el caso de texto al utilizar RNN se puede tener buenos resultados de macr-averaged accuracy para la detección de emociones con pre-procesamientos simples al texto y sin necesidad de uso de Lexicons adicionales. Asimismo, se puede comprobar que los modelos evaluados LSTM y BiLSTM no se diferencian mucho en la cantidad de inferencias por segundo; 15 y 16 por segundo respectivamente. Asimismo, la diferencia entre las métricas de 1% aproximadamente, siendo ambas opciones viables para un trabajo en tiempo real, si es que se requiere valores menores a 20 inferencias por segundo.
- Segundo, se concluye que en el caso de rostros el modelo miniXception supera considerablemente a las opciones pre-entrenadas de VGG16 y ResNet50, esto se evidencia sobre todo en la métrica de 'Recall' donde el modelo miniXception supero por más de 10% a los otros sin importar el tamaño de la imagen. Adicionalmente, este modelo llega a valores alrededor de 20fps siendo muy buena opción para aplicaciones en tiempo real.
- Finalmente, se puede concluir que la detección de emociones en rostros es un problema complejo, debido a que las mismas confusiones entre las mismas expresiones (ej. Ira-Disgusto, Miedo-Sorpresa, Alegría-Disgusto) se presentan en los distintos modelos entrenados.

BIBLIOGRAFÍA

Arriaga, O., Valdenegro-Toro, M., & Plöger, P. G. (2017). Real-time convolutional neural networks for emotion and gender classification, 221–226.

Bakhshi, A., Chalup, S., & Noman, N. (2020). Fast Evolution of CNN Architecture for Image Classification. In *Natural Computing Series* (pp. 209–229). https://doi.org/10.1007/978-981-15-3685-4_8

Becker, C., Kopp, S., & Wachsmuth, I. (2007). Why Emotions should be Integrated into Conversational Agents. *Conversational Informatics: An Engineering Approach*, 49–67. https://doi.org/10.1002/9780470512470.ch3

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, *5*, 135–146. https://doi.org/10.1162/tacl_a_00051

Brandtzaeg, P. B., & Følstad, A. (2017). Why people use chatbots. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *10673 LNCS*, 377–392. https://doi.org/10.1007/978-3-319-70284-1_30

Cabanac, M. (2016). What is emotion?, *6357*(December), 69–83.

Canales, L., Martinez-Ba, P., & Rco. (2018). Emotion detection from text: a survey. *Social Network Analysis and Mining*, *8*(1), 1–8. https://doi.org/10.1007/s13278-018-0505-2

Canziani, A., Paszke, A., & Culurciello, E. (2016). An Analysis of Deep Neural Network Models for Practical Applications, 1–7. Retrieved from http://arxiv.org/abs/1605.07678

Catania, F., Fisicaro, D., Spitale, M., & Garzotto, F. (2019). CORK: A conversational agent framework exploiting both rational and emotional intelligence. *CEUR Workshop Proceedings*, *2327*.

Chatterjee, A., Gupta, U., Chinnakotla, M. K., Srikanth, R., Galley, M., & Agrawal, P. (2019). Understanding Emotions in Text Using Deep Learning and Big Data. *Computers in Human Behavior*, *93*, 309–317. https://doi.org/10.1016/j.chb.2018.12.029

Colah's Blog. (n.d.). Understanding LSTM Networks. Retrieved November 1, 2020, from https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Cutts, M. (2013). *Oxford Guide to Plain English*.

Dhankhar, P. (2019). ResNet-50 and VGG-16 for recognizing Facial Emotions. *International Journal of Innovations in Engineering and Technology (IJIET)*, *13*(4), 126–130.

Ekman, P., Friesen, W. V., & Ellsworth, P. (1973). *Emotion in the Human Face: Guidelines for Research and an Integration of Findings* (Vol. 122). Pergamon Press Inc. https://doi.org/10.1192/bjp.122.1.108

Fan, Y., Lam, J. C. K., & Li, V. O. K. (2018). Video-based emotion recognition using deeply-supervised neural networks. *ICMI 2018 - Proceedings of the 2018 International Conference on Multimodal Interaction*, 584–588. https://doi.org/10.1145/3242969.3264978

Fitzpatrick, K. K., Darcy, A., & Vierhile, M. (2017). Delivering Cognitive Behavior Therapy to Young Adults With Symptoms of Depression and Anxiety Using a Fully Automated Conversational Agent (Woebot): A Randomized Controlled Trial. *JMIR Mental Health*, *4*(2), e19. https://doi.org/10.2196/mental.7785

Gholipour Shahraki, A. (2015). Emotion Mining from Text. *Thesis*.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning: Machine Learning Book, 785. Retrieved from http://www.deeplearningbook.org/

Griol, D., Sanchis, A., Molina, J. M., & Callejas, Z. (2019). Developing enhanced conversational agents for social virtual worlds. *Neurocomputing*, *354*(2019), 27–40. https://doi.org/10.1016/j.neucom.2018.09.099

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. https://doi.org/10.1002/chin.200650130

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory, *1780*, 1735–1780.

Holotescu, C. (2016). MOOCBuddy: a chatbot for personalized learning with MOOCs. *Rochi – International Conference on Human-Computer Interaction*, *8*,

91–94. Retrieved from www.matrixrom.ro

Home | TheySay. (n.d.). Retrieved April 23, 2020, from http://www.theysay.io/

Huang, C., & Zaïane, O. R. (2019). Generating responses expressing emotion in an open-domain dialogue system. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *11551 LNCS*, 100–112. https://doi.org/10.1007/978-3-030-17705-8_9

Hussain, S., Sianaki, O. A., & Ababneh, N. (2019). A Survey on Conversational Agents/Chatbots Classification and Design Techniques, *927*(October), 946–956. https://doi.org/10.1007/978-3-030-15035-8

Hyken, S. (2017, January 7). Ten Customer Service And Customer Experience Trends For 2017. Retrieved April 21, 2020, from https://www.forbes.com/sites/shephyken/2017/01/07/10-customer-service-and-customer-experience-cx-trends-for-2017/

Jha, U., Khant, K., Kotadiya, M., Gamdha, K., & Kansagra, Z. (2019). To Alleviate Depression by Interactive Artificial Conversation Entity. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, *5*(2), 1039–1039. https://doi.org/10.32628/cseit1952285

Joshi, N. (2020, February 23). Choosing Between Rule-Based Bots And AI Bots. Retrieved April 22, 2020, from https://www.forbes.com/sites/cognitiveworld/2020/02/23/choosing-between-rule-based-bots-and-ai-bots/

Kataria, P., Rode, K., Jain, A., Dwivedi, P., & Bhingarkar, S. (2018). User Adaptive Chatbot for Mitigating Depression. *International Journal of Pure and Applied Mathematics*, *118*(16), 349–361.

Krizhevsky, A. (2009). *Learning Multiple Layers of Features from Tiny Images*.

Lecun, Y., Bottou, L., Bengio, Y., & Ha, P. (1998). Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*, (November), 1–46.

Lee, D., Oh, K. J., & Choi, H. J. (2017). The chatbot feels you - A counseling service using emotional response generation. *2017 IEEE International Conference on Big Data and Smart Computing, BigComp 2017*, 437–440.

https://doi.org/10.1109/BIGCOMP.2017.7881752

Lester, J., Branting, K., & Mott, B. (2004). Conversational agents: The practical handbook of internet computing. *Practical Handbook of Internet Computing*, 220–240.

Lipton, Z. C., Berkowitz, J., & Elkan, C. (2015). A Critical Review of Recurrent Neural Networks for Sequence Learning, 1–38. Retrieved from http://arxiv.org/abs/1506.00019

Lison, P., & Tiedemann, J. (2016). OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles. *Proceedings of the 10th International Conference on Language Resources and Evaluation, LREC 2016*, 923–929.

Lommatzsch, A., & Katins, J. (2019). An information retrieval-based approach for building intuitive chatbots for large knowledge bases. *CEUR Workshop Proceedings*, *2454*.

Mahoor, M. (n.d.). AffectNet. Retrieved November 28, 2019, from http://mohammadmahoor.com/affectnet/

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space, 1–12. Retrieved from http://arxiv.org/abs/1301.3781

Mollahosseini, A., Abdollahi, H., & Mahoor, M. H. (2018). Studying Effects of Incorporating Automated Affect Perception with Spoken Dialog in Social Robots. *RO-MAN 2018 - 27th IEEE International Symposium on Robot and Human Interactive Communication*, 783–789. https://doi.org/10.1109/ROMAN.2018.8525777

Mollahosseini, A., Hasani, B., & Mahoor, M. H. (2019). AffectNet: A Database for Facial Expression, Valence, and Arousal Computing in the Wild. *IEEE Transactions on Affective Computing*, *10*(1), 18–31. https://doi.org/10.1109/TAFFC.2017.2740923

Mundra, S., Sen, A., Sinha, M., Mannarswamy, S., Dandapat, S., & Roy, S. (2017). Fine-Grained Emotion Detection in Contact Center Chat Utterances. In *The Pacific-Asia Conference on Knowledge Discovery and Data Mining* (Vol. 10235 LNAI, pp. 337–349). https://doi.org/10.1007/978-3-319-57529-2

Oh, K. J., Lee, D., Ko, B., & Choi, H. J. (2017). A chatbot for psychiatric counseling in mental healthcare service based on emotional dialogue analysis and sentence generation. *Proceedings - 18th IEEE International Conference on Mobile Data Management, MDM 2017*. https://doi.org/10.1109/MDM.2017.64

Plutchik, R. (1982). A psychoevolutionary theory of emotions. *Social Science Information*, *21*(4–5), 529–553. https://doi.org/10.1177/053901882021004003

Posner, J., & Russell, J. A. (2005). The circumplex model of affect : An integrative approach to affective neuroscience , cognitive development , and psychopathology, 715–734.

Poushter, J. (2016). *Smartphone Ownership and Internet Usage Continues to Climb in Emerging Economies*. *Pew Research Center*. https://doi.org/10.1017/CBO9781107415324.004

Python Software Foundation. (n.d.). re — Regular expression operations — Python 3.9.0 documentation. Retrieved November 2, 2020, from https://docs.python.org/3/library/re.html

Q°emotion - Home | Q°emotion - Emotional analysis thanks to AI. (n.d.). Retrieved April 23, 2020, from https://www.qemotion.com/en/

Rehurek, R., & Sojka, P. (2010). Software Framework for Topic Modelling with Large Corpora. *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, 45–50.

Rostylsav Neskorozhenyi. (n.d.). Word embeddings in 2020. Review with code examples | by Rostyslav Neskorozhenyi | Towards Data Science. Retrieved November 7, 2020, from https://towardsdatascience.com/word-embeddings-in-2020-review-with-code-examples-11eb39a1ee6d

Scherer, K. (2000). Psychological Models of Emotion. In J. C. Borod (Ed.), *The Neuropsychology of Emotion* (pp. 137–163). Oxford University Press.

Schuster, M., & Paliwal, K. K. (1997). Bidirectional Recurrent Neural Networks. *IEEE TRANSACTIONS ON SIGNAL PROCESSING*, *45*(11), 2673–2681. https://doi.org/10.1016/s1634-6939(13)59289-1

Scikit-learn. (n.d.). scikit-learn: machine learning in Python — scikit-learn 0.23.2 documentation. Retrieved November 7, 2020, from https://scikit-learn.org/stable/

Sharma, B., Puri, H., & Rawat, D. (2018). Digital Psychiatry - Curbing Depression using Therapy Chatbot and Depression Analysis. *Proceedings of the International Conference on Inventive Communication and Computational Technologies, ICICCT 2018*, (Icicct), 627–631. https://doi.org/10.1109/ICICCT.2018.8472986

Shen, D., Wang, G., Wang, W., Min, M. R., Su, Q., Zhang, Y., … Carin, L. (2018). Baseline needs more love: On simple word-embedding-based models and associated pooling mechanisms. *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, *1*, 440–450. https://doi.org/10.18653/v1/p18-1041

Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 1–14.

Strauss, M., Reynolds, C., Hughes, S., Park, K., McDarby, G., Picard, R., … Tan, T. (2005). Affective Computing and Intelligent Interaction, *3784*(October 2005), 699–706. https://doi.org/10.1007/11573548

Sun, X., Chen, X., Pei, Z., & Ren, F. (2018). Emotional Human Machine Conversation Generation Based on SeqGAN. *2018 1st Asian Conference on Affective Computing and Intelligent Interaction, ACII Asia 2018*, 1–6. https://doi.org/10.1109/ACIIAsia.2018.8470388

Swanson, K., Yu, L., Fox, C., Wohlwend, J., & Lei, T. (2019). Building a Production Model for Retrieval-Based Chatbots, 32–41. https://doi.org/10.18653/v1/w19-4104

TensorFlow. (n.d.-a). TensorFlow. Retrieved April 24, 2021, from https://www.tensorflow.org/

TensorFlow. (n.d.-b). tf.keras.preprocessing.image.ImageDataGenerator. Retrieved April 25, 2021, from https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/image/ImageDataGenerator

Van Eeuwen, M. (2017). Mobile conversational commerce: messenger chatbots as the next interface between businesses and consumers. *University of Twente*, 15.

Varoquaux, G., Buitinck, L., Louppe, G., Grisel, O., Pedregosa, F., & Mueller, A. (2015). Scikit-learn. *GetMobile: Mobile Computing and Communications*, *19*(1), 29–33. https://doi.org/10.1145/2786984.2786995

Velasco-Montero, D., Fernández-Berni, J., Carmona-Galán, R., & Rodríguez-Vázquez, Á. (2018). Performance analysis of real-time DNN inference on Raspberry Pi, 14. https://doi.org/10.1117/12.2309763

Watson Tone Analyzer. (n.d.). Retrieved April 23, 2020, from https://www.ibm.com/watson/services/tone-analyzer/

What Are Word Embeddings for Text? (n.d.). Retrieved November 2, 2020, from https://machinelearningmastery.com/what-are-word-embeddings/

Wirtz, J., Paluch, S., Gruber, T., Lu, V. N., Patterson, P. G., Martins, A., & Kunz, W. H. (2018). Brave new world: service robots in the frontline. *Journal of Service Management*, *29*(5), 907–931. https://doi.org/10.1108/josm-04-2018-0119

Xu, A., Liu, Z., Guo, Y., Sinha, V., & Akkiraju, R. (2017). A new chatbot for customer service on social media. *Conference on Human Factors in Computing Systems - Proceedings*, *2017-May*, 3506–3510. https://doi.org/10.1145/3025453.3025496