People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
University of Ferhat Abbas Sétif-1
Faculty of Sciences
Department of Computer Science

---

**Master's Thesis**
(Mémoire de Fin d'Études)

# Real-time Speech Emotion Recognition of Spoken Algerian Arabic Dialect using Deep Learning and Mobile Deployment




---

**Submitted by**

**Sefiane Ghaith**
**Belamri Mounib**

**Supervisor:** Dr. Hamdi Skander

**Specialty:** RSD — Computer Science
Academic Year: 2025–2026

# Abstract

Algerian Arabic is spoken by over 44 million people yet remains largely absent from speech emotion recognition research. This work presents a complete real-time Speech Emotion Recognition (SER) pipeline for spoken Algerian Arabic dialect, classifying utterances into *angry*, *happy*, *neutral*, and *sad* directly from raw audio. A CNN trained on three-channel MFCC features (static, delta, delta-delta) was built on a labeled corpus of 2,402 recordings using Python, TensorFlow, Librosa, and Flutter.

Test accuracy improved through four stages: baseline CNN (70.95%), residual CNN with SE attention and training-only augmentation (86.72%), a five-model ensemble (88.80%), and knowledge distillation to a compact student (**91.29%**, macro F1 0.8938). The distilled model was converted to INT8 TensorFlow Lite (4.1 MB, 28 ms per window). A Flutter application runs chained on-device preprocessing and inference fully offline, with end-to-end latency of approximately 120 ms on a mid-range Android device. To our knowledge, this is the first published end-to-end real-time Algerian dialect SER system deployed on mobile hardware.

As a complementary contribution, a server-side extended experiment was conducted using a fine-tuned `facebook/wav2vec2-large-xlsr-53` model, a self-supervised speech representation pretrained on 53 languages including Arabic. Fine-tuned on the same Algerian dialect dataset with inverse-frequency class weighting and a 2.5× manual boost for the minority happy class, the wav2vec2 model achieved a test accuracy of **94.61%** (validation accuracy: 95.42%), outperforming the CNN pipeline by 3.32 percentage points overall and raising happy class recall from 79.17% to 87.50%. The model is served through a FastAPI endpoint and integrated into the Flutter application as an optional online inference mode, complementing the on-device CNN with a higher-accuracy cloud-based alternative when network connectivity is available.

**Keywords:** speech emotion recognition, Algerian Arabic dialect, residual convolutional neural network, squeeze-and-excitation attention, MFCC, delta features, data augmentation, ensemble learning, knowledge distillation, TensorFlow Lite, Flutter, real-time inference, low-resource speech.

# Résumé

Le dialecte arabe algérien, parlé par plus de 44 millions de personnes, reste largement absent de la recherche en reconnaissance des émotions dans la parole. Ce mémoire présente un pipeline complet de reconnaissance en temps réel pour la parole en dialecte algérien, classifiant les énoncés en *colère*, *joie*, *neutralité* et *tristesse* à partir de l'audio brut. Un CNN entraîné sur des caractéristiques MFCC à trois canaux (MFCC, delta, delta-delta) a été développé sur un corpus de 2 402 enregistrements annotés à l'aide de Python, TensorFlow, Librosa et Flutter.

La précision progresse en quatre étapes : CNN de référence (70,95 %), CNN résiduel avec attention SE et augmentation (86,72 %), ensemble de cinq modèles (88,80 %), puis distillation vers un étudiant compact (**91,29 %**, F1 macro 0,8938). Le modèle distillé est converti en TensorFlow Lite INT8 (4,1 Mo, 28 ms par fenêtre). Une application Flutter exécute le prétraitement et l'inférence sur appareil via deux interpréteurs TFLite en chaîne, entièrement hors ligne, avec une latence de bout en bout d'environ 120 ms sur un appareil Android milieu de gamme. À notre connaissance, il s'agit de la première démonstration publiée d'un système SER algérien en temps réel déployé sur mobile.

À titre de contribution complémentaire, une expérience étendue côté serveur a été menée en utilisant un modèle `facebook/wav2vec2-large-xlsr-53` affiné sur le même corpus de dialecte algérien. Le modèle wav2vec2, préentraîné sur 56 000 heures de parole multilingue incluant l'arabe, a été affiné avec une pondération inverse des fréquences de classe et un boost manuel de 2,5× appliqué à la classe *joie* — la classe minoritaire la plus difficile. Le modèle a atteint une précision de test de **94,61 %** (précision de validation : 95,42 %), surpassant le pipeline CNN de 3,32 points de pourcentage et portant le rappel de la classe *joie* de 79,17 % à 87,50 %. Le modèle est servi via un point d'accès FastAPI local et intégré dans l'application Flutter en tant que mode d'inférence en ligne optionnel, activé automatiquement lorsque la connexion réseau est disponible et sélectionné par l'utilisateur via un interrupteur à bascule.

**Mots-clés :** reconnaissance des émotions dans la parole, dialecte arabe algérien, réseau de neurones convolutif résiduel, attention Squeeze-and-Excitation, MFCC, caractéristiques delta, augmentation de données, apprentissage par ensemble, distillation de connaissances, TensorFlow Lite, Flutter, inférence en temps réel, traitement de la parole à faibles ressources.

# الملخّص

يُتحدَّث اللهجة العربية الجزائرية من قِبَل أكثر من 44 مليون نسمة، ومع ذلك فإنها ما تزال غائبة إلى حدٍّ كبير عن أبحاث التعرّف على المشاعر في الكلام. يقدّم هذا العمل خطًّا متكاملًا للتعرّف على المشاعر في الكلام باللهجة الجزائرية المنطوقة في الزمن الحقيقي، حيث تُصنَّف العبارات الصوتية إلى أربع فئات: الغضب، الفرح، الحياد، والحزن، انطلاقًا من الصوت الخام مباشرة. بُنيت شبكة عصبية تلافية (CNN) على ميزات MFCC ثلاثية القنوات (ثابتة، دلتا، دلتا-دلتا) باستخدام مُجمّع مُوسَّع مُوسَّم يضم 2,402 تسجيلًا، بأدوات Python وTensorFlow وLibrosa وFlutter.

تحسّنت دقة الاختبار عبر أربع مراحل: CNN أساسية (70,95%)، CNN متبقية مع انتباه SE وتكبير بيانات التدريب فقط (86,72%)، مجموعة من خمس نماذج (88,80%)، ثم تقطير المعرفة نحو نموذج طالب مُدمج (**91,29%**، F1 كلّي 0,8938). حُوّل النموذج المُقطَّر إلى Lite TensorFlow بصيغة INT8 (4,1 ميغابايت، 28 ملي ثانية لكل نافذة). يُشغّل تطبيق Flutter سلسلة مُعالجة مسبقة واستدلالًا على الجهاز دون اتصال، بزمن استجابة من طرف لآخر يقارب 120 ملي ثانية على هاتف Android متوسط المواصفات. على حدّ علمنا، هذا أوّل نظام مُنشر للتعرّف على المشاعر في الكلام باللهجة الجزائرية في الزمن الحقيقي على عتاد محمول.

وبمساهمة تكميلية، أُجريت تجربة موسّعة على الخادم باستخدام نموذج facebook/wav2vec2-large-xlsr-53 المُخصَّص على اللهجة الجزائرية نفسها. النموذج المُدرَّب ذاتيًا مسبقًا على 56,000 ساعة من الكلام متعدد اللغات بما فيها العربية، وُسِّع بوزن عكسي لتكرار الفئات وتعزيز يدوي بمقدار $2{,}5\times$ لفئة الفرح --- الأقل تمثيلًا. بلغت دقة الاختبار **94,61%** (والتحقق 95,42%)، متفوقة على خط CNN بـ 3,32 نقطة مئوية، ورفعت استرجاع فئة الفرح من 79,17% إلى 87,50%. يُقدَّم النموذج عبر واجهة FastAPI ويُدمَج في تطبيق Flutter كوضع استدلال اختياري عند توفر الشبكة.

**الكلمات المفتاحية:** التعرّف على المشاعر في الكلام، اللهجة العربية الجزائرية، شبكة عصبية تلافية متبقية، انتباه Squeeze-and-Excitation، MFCC، ميزات دلتا، تكبير البيانات، التعلم بالمجموعات، تقطير المعرفة، TensorFlow Lite، Flutter، الاستدلال في الزمن الحقيقي، معالجة الكلام منخفضة الموارد.

# Acknowledgments


## بسم الله الرحمن الرحيم

الحمد لله الذي وفّقنا وأعاننا على إتمام هذا العمل.

نتقدم بجزيل الشكر وعميق الامتنان إلى الأستاذ المشرف الدكتور حمدي سكندر، الذي لم يكن مجرد موجّه أكاديمي، بل كان حاضرًا في اللحظات الأصعب. حين واجهنا نتيجة 70,95 بالمئة في نموذجنا الأول ولم نكن نعرف السبب الحقيقي، كان هو من وجّهنا نحو المكان الصحيح، ودفعنا نحو فهم أعمق للبنية والبيانات معًا. هذا التوجيه غيّر مسار العمل بأكمله.

نشكر أعضاء لجنة المناقشة، رئيسها والمناقشين، على الوقت الذي يمنحونه لقراءة هذا العمل والمشاركة في مناقشته. نأمل أن يكون هذه المذكرة في مستوى ذلك الاهتمام.

نشكر عائلاتنا على أشياء محددة: على الصبر في الأوقات التي كان فيها هذا المشروع يستحوذ على كل شيء، وعلى الثقة التي أبدوها حتى حين كنا نحن أنفسنا في شك، وعلى الحضور الهادئ الذي لا يُعوَّض.

وأخيرًا نشكر بعضنا البعض. خمس سنوات من الدراسة المشتركة، ومذكرة بُنيت معًا من البداية حتى النهاية، من النموذج الأول الذي أعطانا 70 بالمئة حتى النظام الأخير الذي يعمل في الوقت الفعلي على هاتف محمول. هذا النوع من الشراكة الحقيقية لا يحدث دائمًا.

# Contents


**1 Introduction** **8**
- 1.1 Research Context and Motivation . . . 8
- 1.2 Problem Statement . . . 9
- 1.3 Research Contributions . . . 10
- 1.4 Structure of the Thesis . . . 11

**2 Background and Theoretical Foundations** **12**
- 2.1 Introduction . . . 12
- 2.2 Speech Signal Processing and Acoustic Feature Extraction . . . 12
  - 2.2.1 From Raw Audio to Numerical Representations . . . 12
  - 2.2.2 Mel-Frequency Cepstral Coefficients (MFCCs) . . . 12
  - 2.2.3 Delta and Delta-Delta Features . . . 13
  - 2.2.4 Data Augmentation for Audio . . . 13
- 2.3 Deep Learning Architectures for Speech Classification . . . 14
  - 2.3.1 Convolutional Neural Networks (CNNs) . . . 14
  - 2.3.2 Residual Connections . . . 14
  - 2.3.3 Squeeze-and-Excitation (SE) Attention . . . 14
  - 2.3.4 LSTM and BiLSTM . . . 15
- 2.4 Ensemble Learning and Knowledge Distillation . . . 15
  - 2.4.1 Ensemble Learning . . . 15
  - 2.4.2 Knowledge Distillation . . . 15
- 2.5 Speech Emotion Recognition: Concepts and Task Definition . . . 15
- 2.6 Mobile Deployment Technologies . . . 16
  - 2.6.1 TensorFlow Lite . . . 16
  - 2.6.2 Flutter . . . 16
- 2.7 Self-Supervised Speech Representations: wav2vec 2.0 . . . 16
- 2.8 Summary . . . 16

**3 Literature Review** **18**
- 3.1 Introduction . . . 18
- 3.2 Speech Emotion Recognition . . . 18
  - 3.2.1 Deep Learning for Speech Classification . . . 18
  - 3.2.2 The Critical Role of Data Augmentation . . . 19
  - 3.2.3 Residual Architectures, Attention, Ensemble Learning, and Knowledge Distillation . . . 20
- 3.3 Arabic Dialect Speech Processing . . . 21
  - 3.3.1 Arabic Speech and Emotion Processing . . . 21
  - 3.3.2 Algerian Dialect: A Particularly Low-Resource Case . . . 21

3.4 Mobile Deployment of Speech Models . . . 22
3.5 Summary of Related Works . . . 23
3.6 Research Gaps and Positioning of This Work . . . 24

**4 Methodology** **27**
4.1 Overview . . . 27
4.2 Dataset Description and Preparation . . . 27
4.2.1 Dataset Overview . . . 27
4.2.2 Dataset Statistics and Class Distribution . . . 28
4.2.3 Data Splitting . . . 28
4.3 Audio Preprocessing and Feature Extraction . . . 28
4.3.1 Preprocessing Pipeline . . . 28
4.3.2 MFCC Feature Extraction and Three-Channel Input . . . 29
4.4 Model Design and Training . . . 30
4.4.1 Baseline CNN Architecture . . . 30
4.4.2 Baseline Results . . . 31
4.4.3 Improved CNN: Residual Connections and SE Attention . . . 31
4.4.4 Training Configuration . . . 32
4.4.5 Diagnosis: Why the Baseline Model Underperformed . . . 33
4.4.6 Four-Stage Improvement Pipeline: From 70.95% to 91.29% . . . 33
4.5 Model Optimization for Mobile Deployment . . . 37
4.5.1 TensorFlow Lite Conversion and Quantization . . . 37
4.6 On-Device Preprocessing Pipeline . . . 38
4.6.1 The Preprocess Consistency Problem . . . 38
4.6.2 Why Dart and Native Reimplementation Were Rejected . . . 38
4.6.3 TFLite Implementation: Replacing Librosa Operation by TFLite Operation . . . 38
4.6.4 Export and Conversion . . . 40
4.6.5 Numerical Validation . . . 41
4.7 Flutter Application Development . . . 42
4.8 Extended Experiment: wav2vec2 Server-Side Model . . . 42
4.8.1 Model and Fine-Tuning Configuration . . . 43
4.8.2 Per-Class Results . . . 43
4.8.3 FastAPI Server Deployment . . . 44
4.8.4 Flutter Integration: Online/Offline Mode . . . 45
4.9 Experimental Evaluation . . . 45
4.9.1 Evaluation Metrics . . . 45
4.9.2 Per-Class Results of the Final Distilled Model . . . 46
4.9.3 Four-Stage Improvement Summary . . . 46
4.9.4 Real-Time System Performance . . . 46

**5 General Conclusion** **47**